Computersystemen 2 Theorie

4. Bestandsbeheer

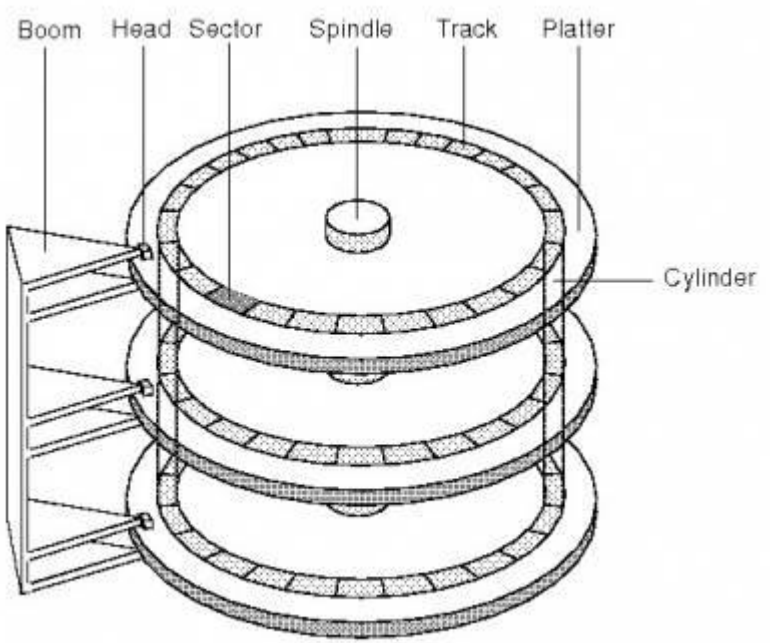
Karel de Grote Hogeschool

10-2024

Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

HDD - Hard Disk Drive



- harde schijf
  - disks
  - heads
  - cilinders
  - tracks
  - sectors
- magnetisch
  - weinig schokbestendig
  - latency (kop heen en weer bewegen)
- bij delete file
  - file wordt verwijderd uit directory tabel
  - sectoren van file blijven op disk staan (en kunnen nadien overschreven worden)

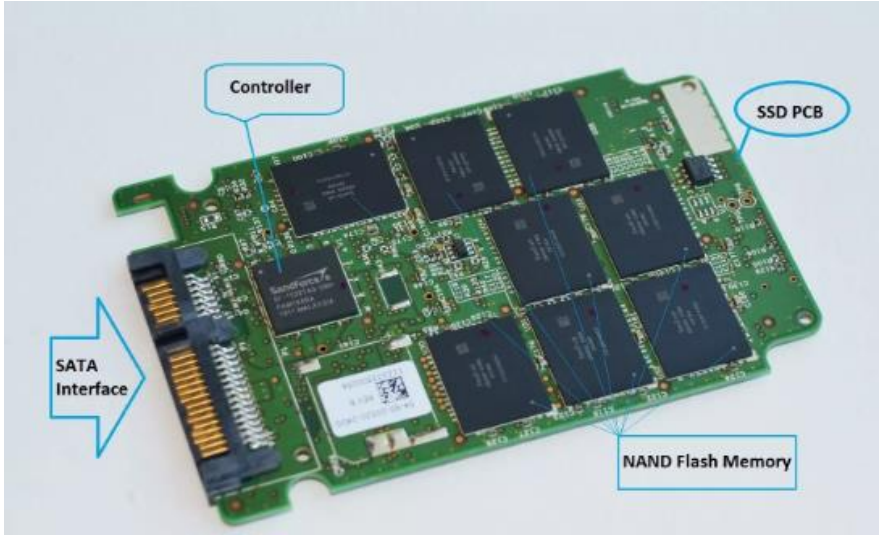
SSD - Solid-state Drive

- SSD:
  - Interface: SATA, …
  - Controller (firmware)
  - NAND Flash Memory:

    vgl. met static RAM (D-flipflop)

    maar met floating gate transitoren

    (bewaren toestand bij power off)



- Voltage levels:
  - SLC (single level cell): 1 bit/cell, 0 of 1 (bv. 0 en 3V)
  - MLC (multi level cell): 2 bits/cell, 00, 01, 10 of 11 (bv. 0, 1, 2 en 3V)
    - hogere capaciteit, trager, minder betrouwbaar
  - zelf TLC (tripe level cell): 3bits/cell

SSD - Solid-state Drive

- geschreven page kan niet overschreven worden
  - eerst leeg maken
  - bij delete file:
    - niet enkel file verwijderen uit directory
    - TRIM instructie aan OS toegevoegd: doorgeven aan SSD om pages te markeren om leeg te maken (gebeurt door de controller van de SSD)
- cellen degraderen bij schrijven
  - aantal write cycles is beperkt
  - geen defragmentatie!
  - wear-leveling
    - erases & writes are evenly distributed over the SSD IC's
    - move frequently accessed data to lower used blocks

SSD - HDD

	SDD	HDD
Access Time	35 to 100 µsec	5 to 10 ms
Price/Capacity	Most Expensive	Cheaper (>1TB)
Reliability	No moving parts (ICs)	Platters
Power	Less	More
Noise	No noise	Spinning Disks Moving heads
Size	Many size + small	Typical only 3,5" and 2,5"
Heat	Almost no heat	Moving parts cause heat
Magnetism	No effect	Erase possible
FileSystem	No Defragmentation	Defragmentation

Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

Schijftoegang - RAID

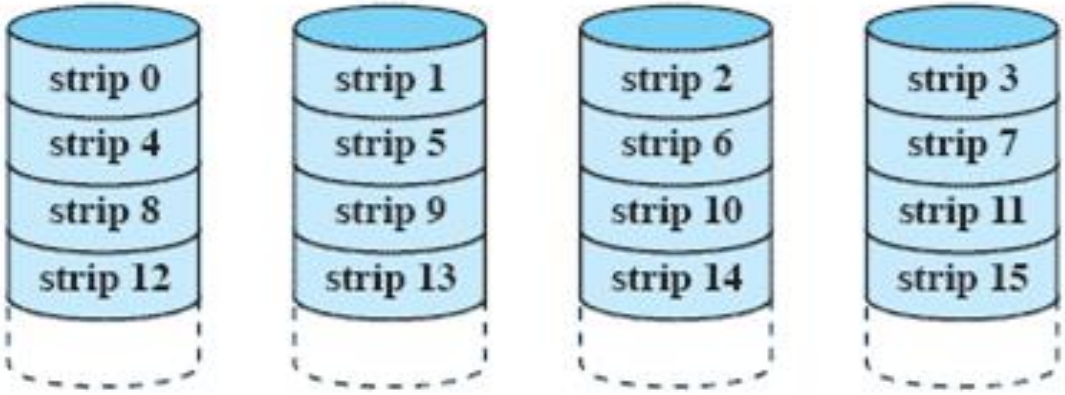
- RAID: Redundant Array of Independent Disks
- virtuele schijf bestaande uit verschillende fysische harde schijven
- filesystemen kunnen verspreid staan over verschillende harde schijven
- voordelen
  - grotere filesystemen
  - meerdere schijfoperaties tegelijk uitvoeren
  - redundantie is mogelijk
- implementatie
  - software: in het OS
  - hardware: OS weet van niets

Schijftoegang - RAID

- RAID heeft verschillende niveau's
  - 0 tot 6
  - enkel 0, 1 en 5 komen veel voor
  - je kan ook combinaties maken (vb: RAID-10, RAID-51)

RAID-O (striping)

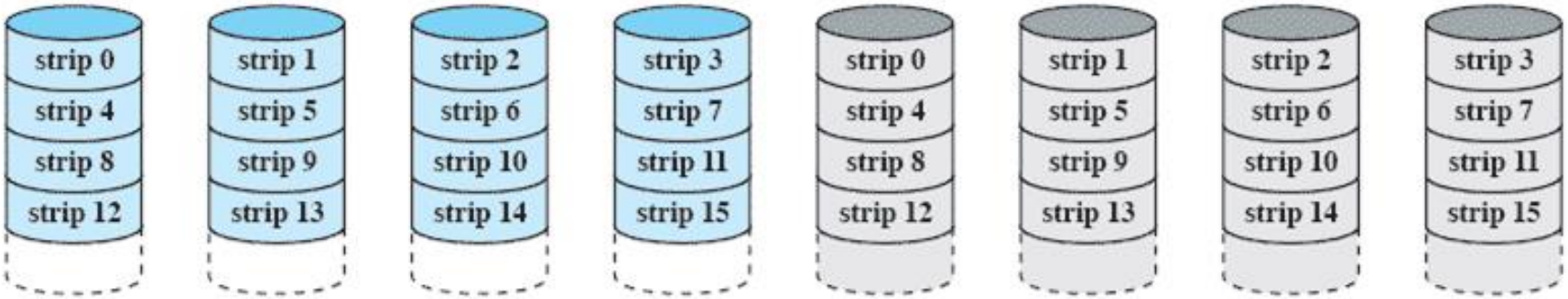
- verdeel de schijven in kleine delen (strips)
- verspreid de data over de schijven



- voor-en nadeel?
  - voordeel: snellere toegang (parallel)
  - nadeel: fout in 1 schijf ==> alle data verloren

RAID-1 (mirroring)

- verdubbel het aantal schijven
- bewaar alle data 2 keer (mirroring)

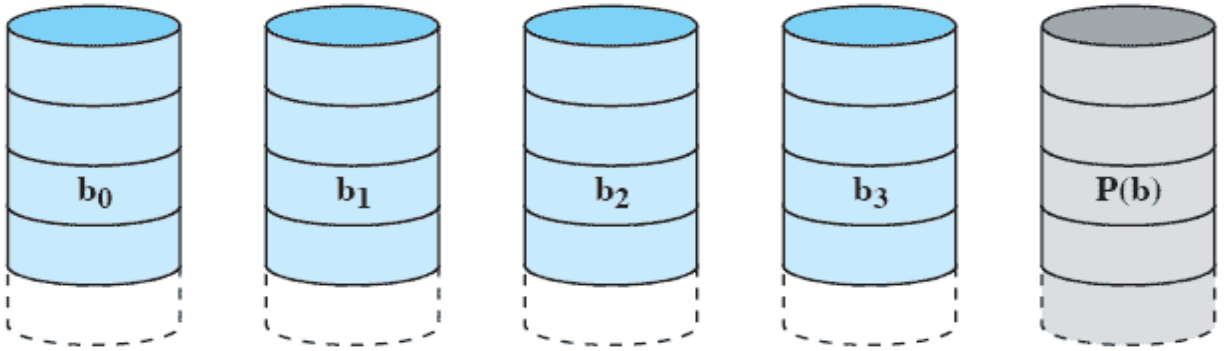


combinatie RAID 0 en 1

- voor- en nadeel?
  - voordeel: foutcorrectie mogelijk, snelle leestoegang
  - nadeel: duur (2 x aantal schijven nodig)

RAID-3

- gebruik 1 schijf voor "pariteitsbits" (redundancy)



- voorbeeld: P(b) = b0 ⊕ b1 ⊕ b2 ⊕ b3 (even pariteit)
- stel: schijf 2 is stuk en er wordt het volgende gelezen:
  - 1010 1101, 1010 1100, xxxx xxxx, 0010 0000, 1001 0101
  - Wat was de waarde van b2?

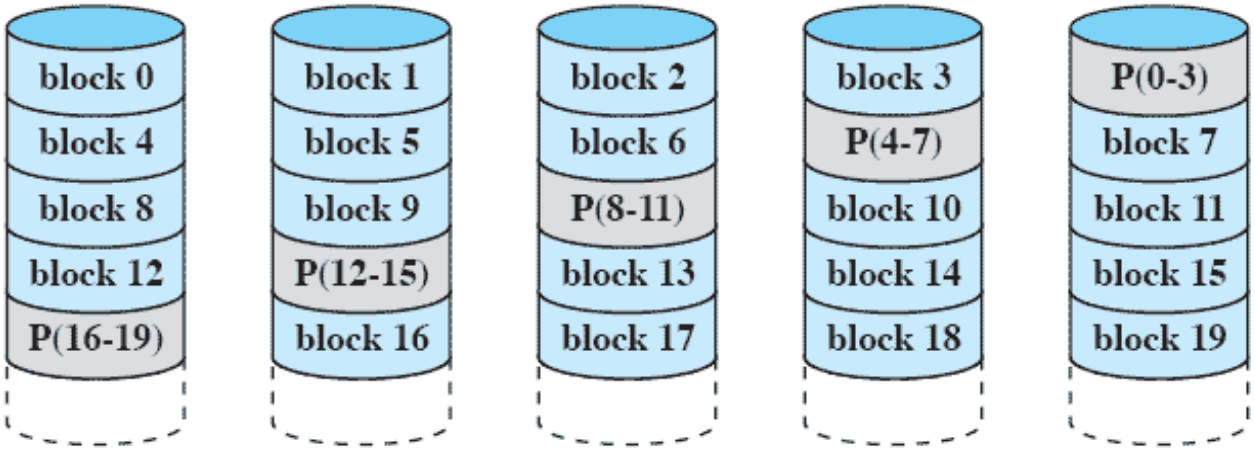
OPDRACHT

RAID-3

- voor- en nadelen?
  - voordeel:
    - goedkoper dan RAID-1
    - mogelijkheid tot correctie
    - snel lezen
  - nadeel:
    - traag bij schrijven: redundancy staat steeds op dezelfde schijf

RAID-5

- de pariteitsblokken worden gespreid over de schijven



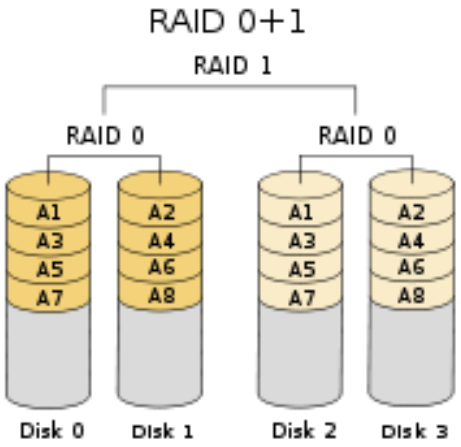
- voordeel: sneller schrijven

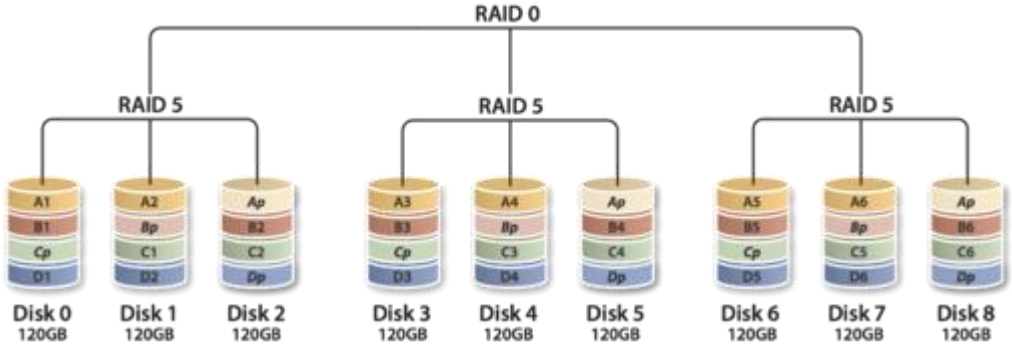
RAID-6

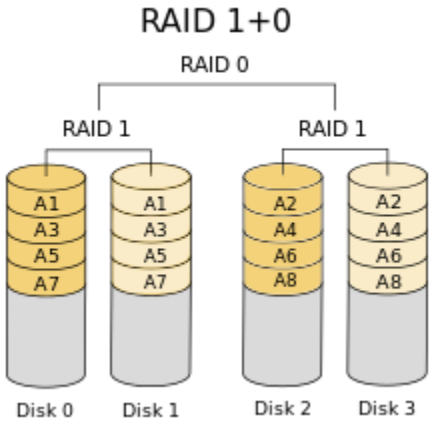
- gebruik meerdere schijven voor redundancy
- nadeel:
  - meer schijven nodig
  - berekeningen worden ingewikkelder (Reed-Solomon codes)
- voordeel: meerdere schijven mogen tegelijk stuk gaan

Nested/Hybrid RAID

- RAID 01, RAID 10, RAID 50





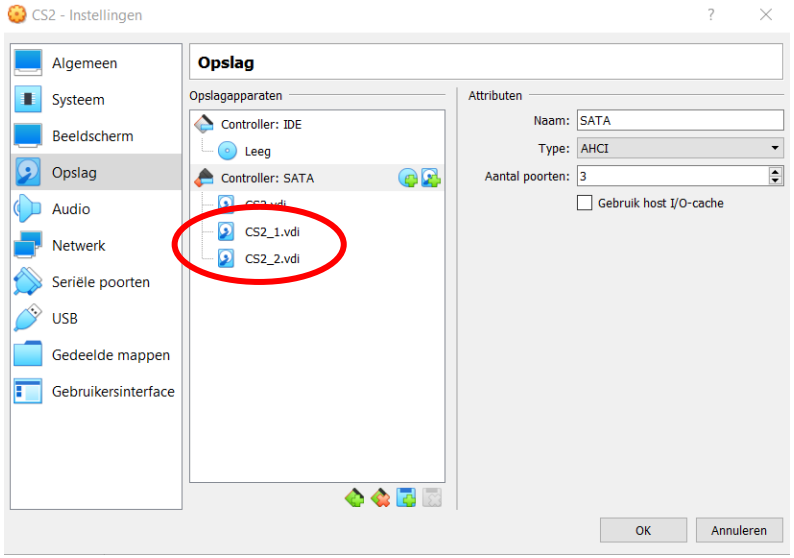


RAID is no substitute for back-up!

- All RAID levels except RAID 0 offer protection from a single drive failure. A RAID 6 system even survives 2 disks dying simultaneously. For complete security you do still need to back-up the data from a RAID system.
- That back-up will come in handy if all drives fail simultaneously because of a power spike.
- It is a safeguard when the storage system gets stolen.
- Back-ups can be kept off-site at a different location. This can come in handy if a natural disaster or fire destroys your workplace.
- The most important reason to back-up multiple generations of data is user error. If someone accidentally deletes some important data and this goes unnoticed for several hours, days or weeks, a good set of back-ups ensure you can still retrieve those files.

RAID lab

1. Voeg 2 (virtuele) disken toe aan je Linux virtuele machine
2. Configureer RAID1: https://www.linuxbabe.com/linux-server/linux-software-raid-1-setup#
   - gebruik GPT partitie tabellen i.p.v. MBR
   - welke raid levels worden ondersteund?
   - simuleer uitvallen van 1 disk



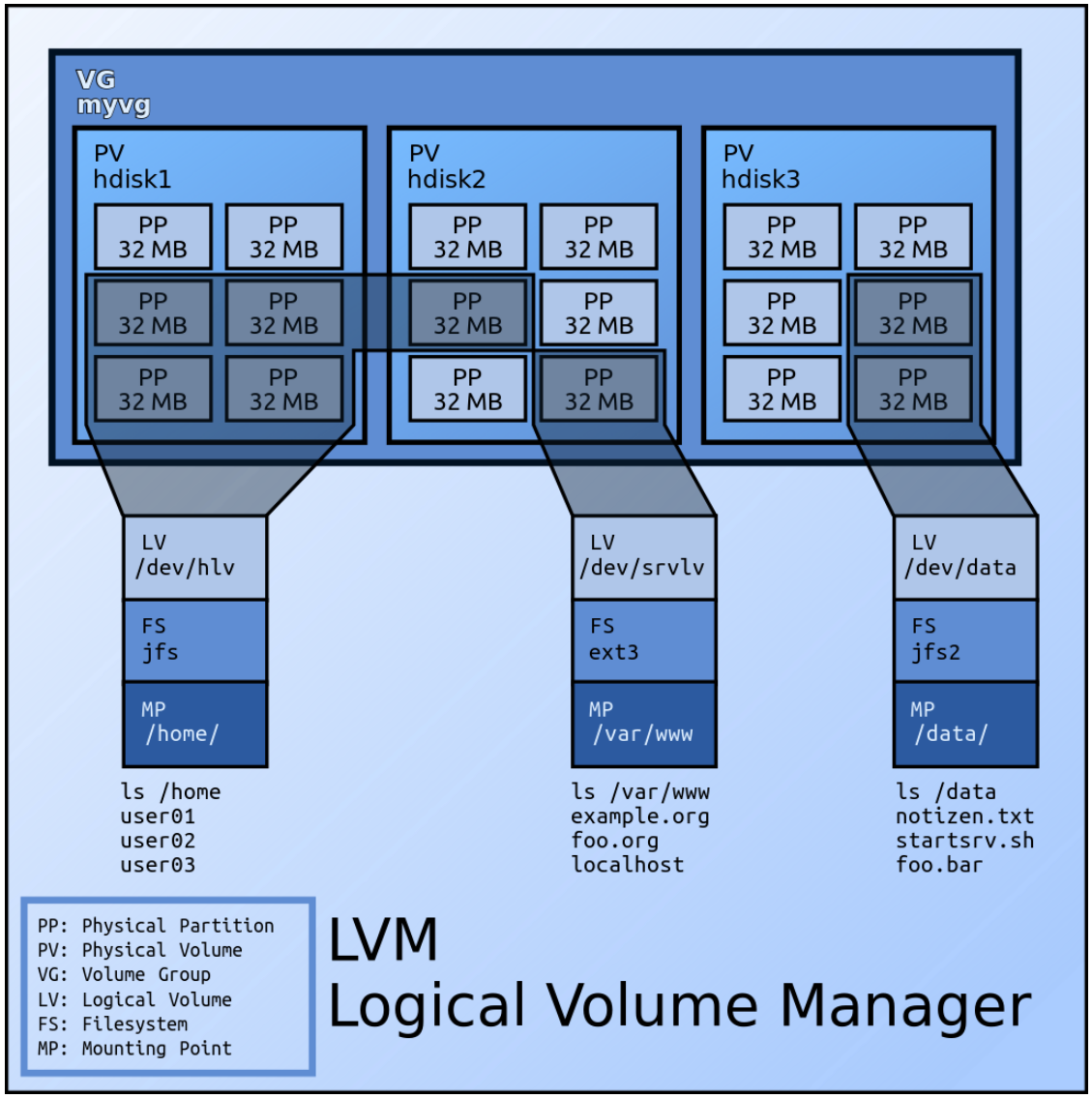
1. Partitioneren
   - fdisk
2. RAID opzetten
   - mdadm --create /dev/md0 --level=mirror --raid-devices=2 /dev/sdb1 /dev/sdc1
   - mdadm --detail /dev/md0
   - mdadm --examine /dev/sdb1 /dev/sdc1
3. Formatteren
   - mkfs.ext4 /dev/md0
4. Mounten
   - mount /dev/md0 /mnt/raid1

OPDRACHT

Logical volume management

- LVM
  - logical volumes (=logische partitie) kunnen meerdere fysische partities bevatten
  - dynamisch resizen van volume
  - snapshots van volumes
  - encryptie van volumes
  - RAID functionaliteit

Logical volume management



Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

File management

- = deel van OS dat boven disk i/o draait
- definieert
  - bestanden
  - directories (folders, mappen)
  - links (shortcuts)
  - vuilbak
  - meta-informatie (file attributes)

Bestandsattributen

- eigenaar van het bestand
- lees/schrijfrechten
  - voorbeeld Unix: rwx rwx rwx
- is bestand gecomprimeerd?
- moet bestand gebackupt worden?
- tijdstip creatie, laatste wijziging, laatste toegang
- met welke applicatie moet dit bestand geopend worden?
- zijn er verschillende versies van dit bestand?
- ...

Interne fragmentatie

- bestanden worden bewaard in blokken op de schijf
- voorbeeld
  - blokken van 4096 bytes *
  - bestandsgrootte van 10 000 bytes
  - 3 blokken nodig
  - laatste blok is niet volledig gevuld
  - = interne fragmentatie

- hoeveel bytes gaan er gemiddeld verloren per bestand?
  - voorbeeld: blokken van 4096 bytes, 1 000 000 bestanden op schijf ==> hoeveel plaats gaat er verloren?
- besluit: maak blokgrootte zo klein mogelijk

*** Blokgrootte is altijd een veelvoud van 512 byte. Waarom? De blokgrootte kan meegegeven worden bij het formatteren, zie man mkfs.ext4**

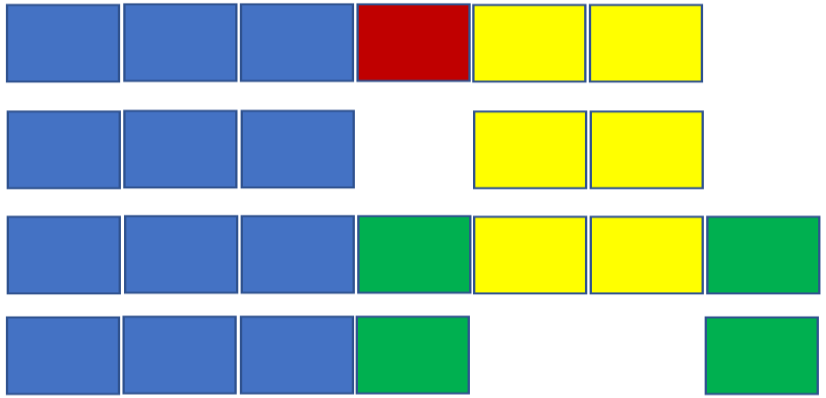
OPDRACHT

Voorbeeld: FAT filesysteem

- FAT16
  - FAT16 kan schijven indelen in hoogstens 65536 blokken
  - blokken kunnen maximaal 65536 bytes groot zijn
  - ==> maximale diskcapaciteit = 4 GiB
  - stel: 10 000 bestanden
  - ==> door interne fragmentatie gaat 327,68 MB verloren
- FAT32
  - heeft zelfde max blokgrootte
  - maar nu 4 miljard blokken mogelijk

(Externe) Fragmentatie

- bestanden worden geschreven en terug verwijderd
  - er ontstaan dus "gaten"
  - bestanden kunnen verspreid zijn over niet-aangrenzende blokken
- sequentieel lezen van bestand: kop HDD heen en weer ==> traag
- beter: grote bestanden in aansluitende blokken
- oplossing: defragmentatie*
- hoe groter de blokgrootte, hoe minder externe fragmentatie
  - vs interne fragmentatie...



*** Niet bij SSD om degradatie cellen te voorkomen. Fragmentatie kan bij SSD veel minder/geen kwaad. Waarom?**

Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

De allocation table

- Er is een soort inhoudstafel nodig voor de bestanden
  - op welke blokken staat het bestand?
  - file allocation table (FAT)

- Twee mogelijkheden:
  - Eén per filesysteem: 1 FAT voor alle bestanden (Windows)
  - Eén per file: inode die allocation table bevat (Linux)

FAT 12, 16, 32

- Schijf/partitie bevat
  - blok met bootsector
    - bootcode
    - gegevens over de schijf
  - FAT: 2 keer (voor foutcorrectie)
  - root folder
  - rest van de disk

FAT 12, 16, 32

- schijf van 131 072 bytes
  - 32 blokken van 4096 bytes
  - 1 partitie
- blok 0 = bevat boot sector
- blok 1 = FAT1
- blok 2 = FAT2
- blok 3 = root folder

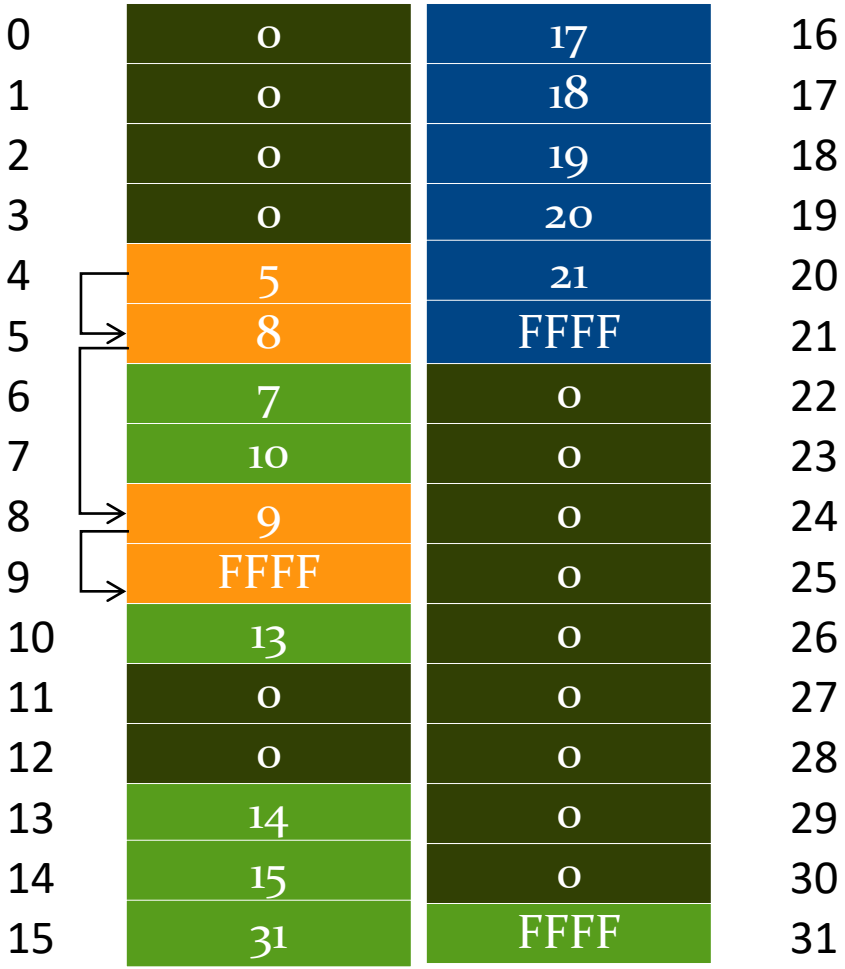
0	16
1	17
2	18
3	19
4	20
5	21
6	22
7	23
8	24
9	25
10	26
11	27
12	28
13	29
14	30
15	31

FAT 12, 16, 32

filename	start	length	attributes
readme.txt	4	15 867	...
myApp.exe	6	27 814	...
verslag.doc	16	22 010	...

root directory tabel: bevindt zich in blok 3

FAT



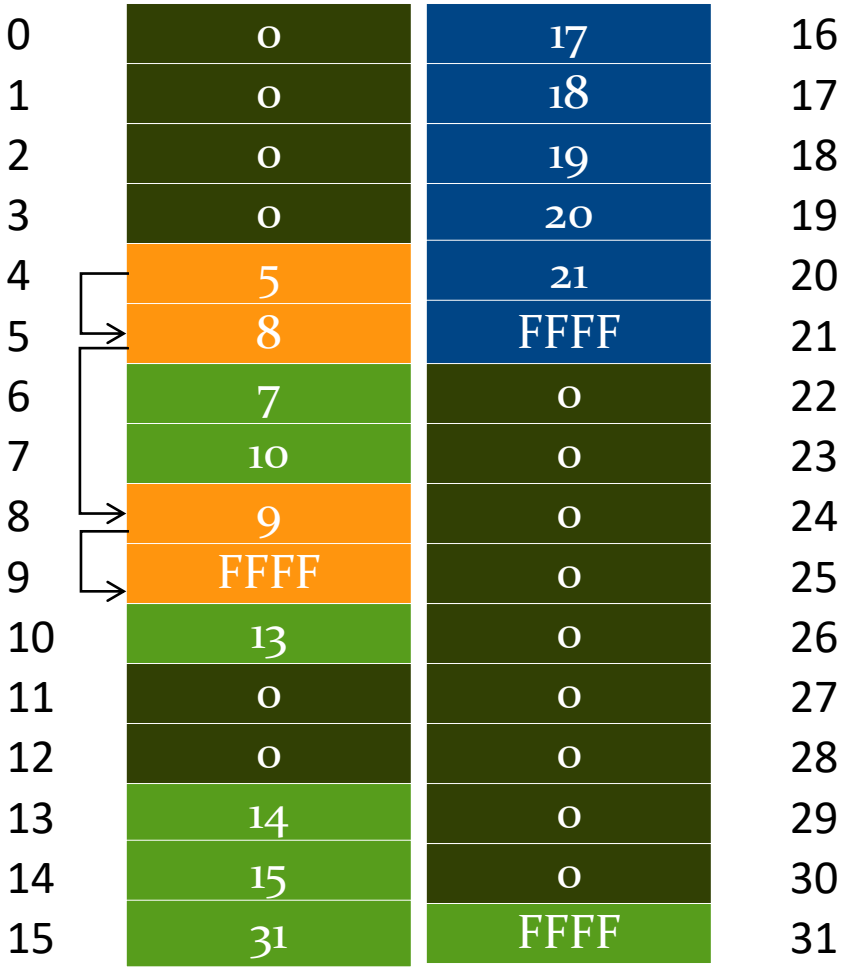
array of int (12, 16, 32)
met evenveel elementen
als blokken in het filesysteem;
bevindt zich in blok 1 en 2

Blokken met nullen worden niet door files gebruikt en zijn dus beschikbaar voor nieuwe files (behalve 1ste vier, waarom?)

FAT voor- en nadelen

- voordeel: heel eenvoudig
- nadeel: heel traag
  - voor iedere blok die geschreven wordt, moet de FAT aangepast worden
  - de FAT moet 2 keer geschreven worden (mirroring)
  - kop moet heel de tijd heen en weer
  - oplossing: caching
    - hou FAT in memory en schrijf af en toe weg naar schijf

FAT oefening



- stel dat het OS een bestand wil bijmaken
  - testje.txt
  - 10.340 bytes
- wat moet het OS doen?

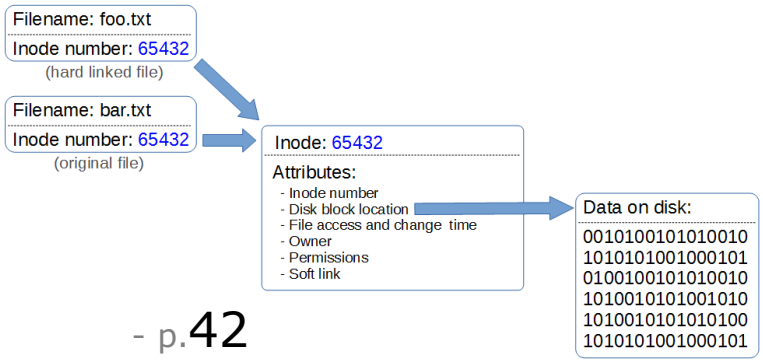
OPDRACHT

Inhoud

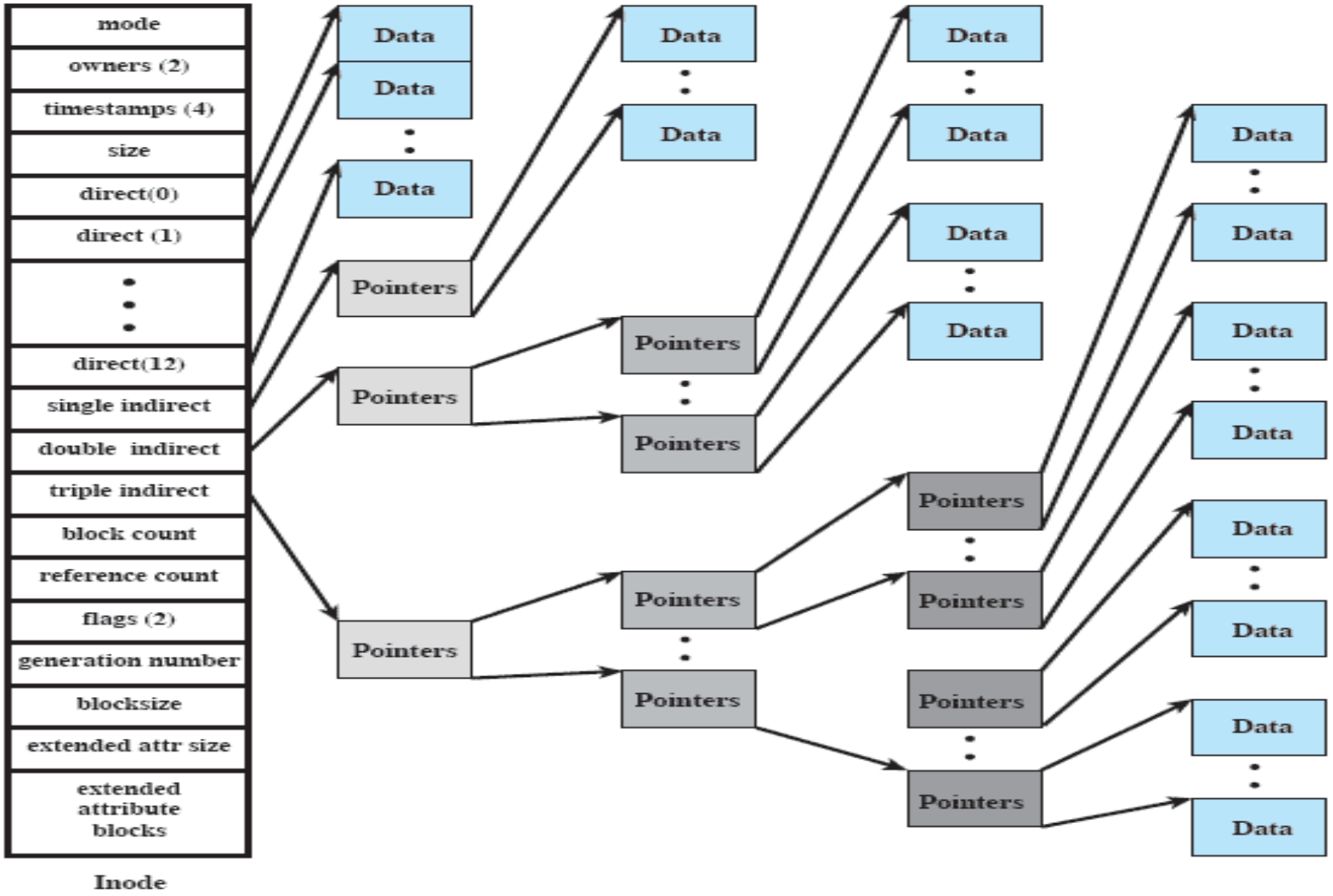
- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

Linux inodes

- Unix heeft een inode per bestand (=datastructuur)
- directory is blok gevuld met pointers naar inodes
- inode bevat filename, lengte, ...
- inode bevat 16 'pointers'
  - eerste 13 wijzen naar de eerste 13 blok van het bestand
  - de 14e verwijst naar een 'single indirect block'. Die bevat pointers naar de volgende sectoren van het bestand
  - de 15e verwijst naar een 'double indirect block'. Die bevat pointers naar single indirect blocks
  - de 16e verwijst naar een 'triple indirect block'. Die bevat pointers naar double indirect blocks



Linux inodes



Inodes

- stel: blokgrootte van 512 bytes, pointers worden opgeslagen als 32 bit unsigned integers
  - blok kan 512/4 = 128 pointers bevatten
- hoe groot kan een bestand worden zonder dat er indirects nodig zijn?
  - 13 * 512
- hoe groot kan een bestand worden zonder dat er double indirects nodig zijn?
  - 13 * 512 + 128 * 512
- hoe groot kan een bestand maximaal worden?
  - $13 * 512 + 128 * 512 + 128^2 * 512 + 128^3 * 512 \approx 1$ GiB

- Oefening: maak de berekening opnieuw voor een blokgrootte van 4096 bytes

Blokgrootte filesysteem

- De blokgrootte van een filesysteem bepaalt:
  - Hoeveelheid interne fragmentatie
  - Impact van externe fragmentatie
  - Maximale grootte van het filesysteem
  - Maximale grootte van een file
  - Performantie van het filesysteem
- Keuze van de blokgrootte:
  - Afwegen van wat belangrijk is
  - Afhankelijk van soort files op het filesysteem
  - Snelheid <-> verlies
  - Maximale grootte file/filesysteem

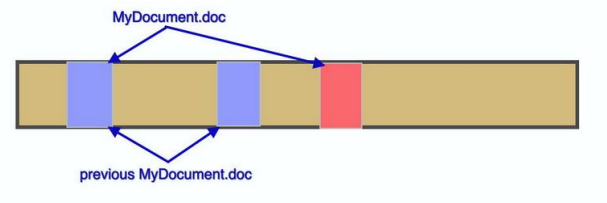
mkfs.ext4 -b *block_size*
mkfs.vfat -s *sectors_per_block*

Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

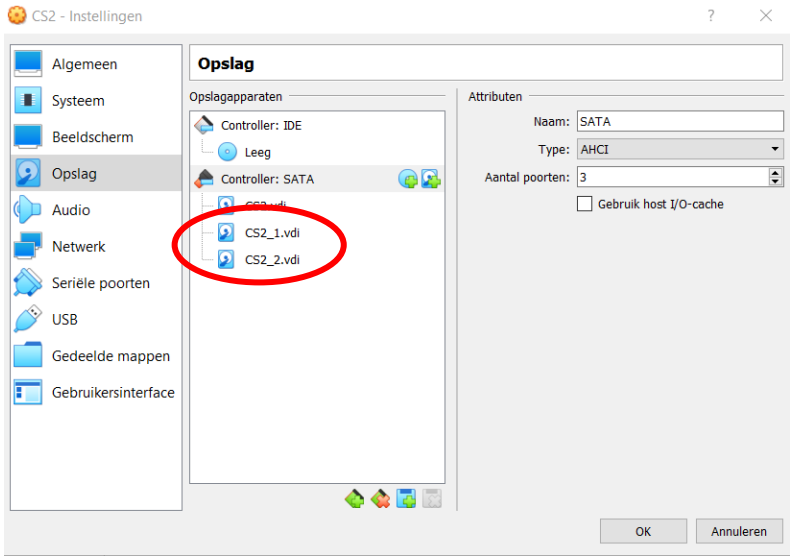
Next gen filesystems

- ZFS: Zettabyte File System
- Btrfs: B-tree FS, Better FS
- combination of filesystem and logical volume manager
  - protection against data corruption
  - high storage capacity
  - copy-on-write
  - snapshots and cloning
  - RAID



ZFS lab

1. Hergebruik de 2 (virtuele) disken je Linux virtuele machine en verwijder de "md raid"
2. Configureer ZFS met dezelfde 2 partities
   https://ubuntu.com/tutorials/setup-zfs-storage-pool




```
zpool create new-pool /dev/sdb1 /dev/sdc1                          # striping
zpool create new-pool mirror /dev/sdb1 /dev/sdc1                   # mirroring
zpool create -m /usr/share/pool new-pool mirror /dev/sdb1 /dev/sdc1 # includes mounting
zpool status
zpool destroy new-pool
```

OPDRACHT

Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

Disken delen over het netwerk

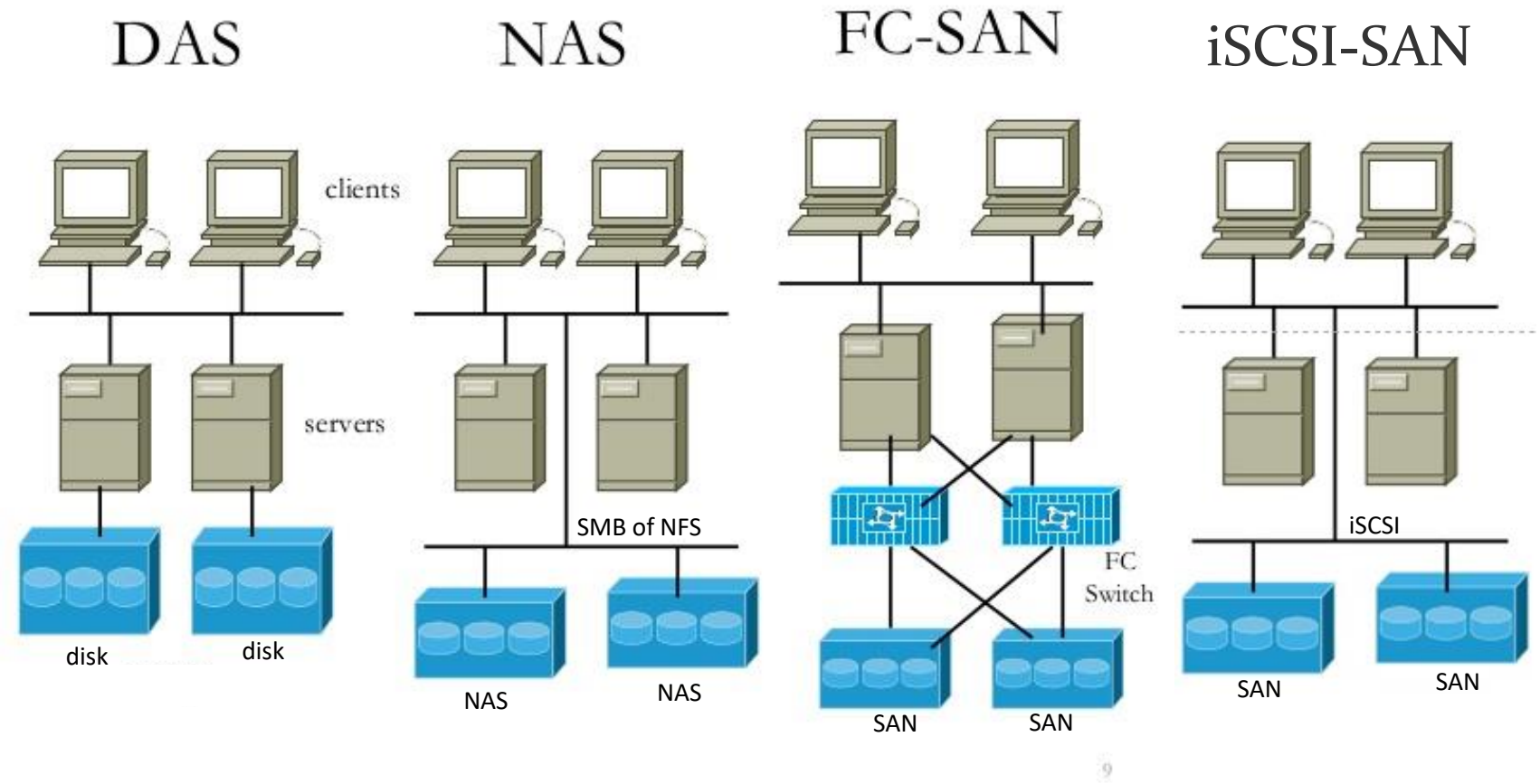
- DAS
  - direct attached storage
  - HDD/SSD zit in computer (of is direct aangesloten aan computer)
- Disken delen over het netwerk
  - NAS: deelt filesystemen over netwerk (“netwerk drives” of “shares”)
  - SAN: deelt blokken over het netwerk

SAN - NAS

- Wat is het verschil tussen SAN en NAS?

	NAS	SAN
afkorting staat voor	Network attached storage	Storage area network
werkt op welk niveau	file-level	block-level
filesysteem staat op	NAS box / File server	Computer
gebruikt welk netwerk	gewone TCP/IP netwerk • SMB (server message block)/CIFS (common Internet filesystem) • NFS (network filesystem)	• Fibre Channel met SAN switches (storage area network) • iSCSI: SCSI in TCP/IP packetten

SAN - NAS


DAS
NAS
FC-SAN
iSCSI-SAN
clients
servers
disk
disk
SMB of NFS
NAS
NAS
FC
Switch
SAN
SAN
iSCSI
SAN
SAN

Inhoud

- HDD en SSD
- RAID
- Bestandsystemen
- FAT Tabel
- Linux Inodes
- Next Gen File Systems
- NAS en SAN
- Herhalingsvragen

Herhalingsvragen

- hoe werkt een HDD? Hoe werkt een SSD?
- wat doet RAID-x? vergelijk RAID-x met RAID-y… Ken de voor- en nadelen.
- Stel: RAID-5 systeem met ... harde schijven (waarvan 1 voor redundantie). Schijf ... gaat kapot. De data op de andere schijven is .... Reconstrueer de data op de kapotte harde schijf

Herhalingsvragen

- wat is caching (bij harde schijven). Wat is het voordeel? Wat is het nadeel?
- wat is interne fragmentatie in een bestandsysteem? geef een voorbeeld.
- Hoeveel ruimte gaat er aan interne fragmentatie verloren op een harde schijf van ... als je weet dat er ... bestanden op staan en de blokgrootte gelijk is aan ... bytes?
- wat is (externe) fragmentatie van een harde schijf? Hoe ontstaat dit?
- wat is een "file allocation table"?

Herhalingsvragen

- gegeven volgende FAT: ... in welke sectoren staat een bestand dat op sector ... begint?
- gegeven volgende FAT: ... Hoeveel files staan hier op?
- gegeven volgende FAT: ... met een blokgrootte van ... hoeveel plaats is er nog vrij voor data op deze schijf?
- gegeven volgende FAT: ... met een blokgrootte van ... Stel dat je een nieuw bestand van ... bytes wil bijmaken. Teken dan de nieuwe FAT.
- gegeven een unix file system met een blokgrootte van ... Wat is de maximale grootte van een bestand?
- gegeven een unix file system met een blokgrootte van ... Wat is de maximale grootte van een bestand als je geen (dubbel/triple) indirects mag gebruiken?

Herhalingsvragen

- Hoe bepaal je de blokgrootte van een filesysteem?
- Wat is LVM?
- Wat doen volgende commando's: fdisk, mdadm, mkfs, mount
- Wat zijn eigenschappen van next gen filesystems?
- Wat doet het zpool commando?
- Wat is het verschil tussen SAN en NAS?