

# ZFS

Siste ord innen filsystemer

Trond Endrestøl

Fagskolen Innlandet, IT-avdelingen

3. januar 2014

- Filene til foredraget er tilgjengelig gjennom:
  - Subversion: `svn co svn://svn.ximalas.info/zfs-foredrag`
  - Web: `svnweb.ximalas.info/zfs-foredrag`
  - Begge metodene er tilgjengelig med både IPv4 og IPv6
- `zfs-foredrag.foredrag.pdf` vises på lerretet
- `zfs-foredrag.handout.pdf` er mye bedre for publikum å se på
- `zfs-foredrag.handout.2on1.pdf` og `zfs-foredrag.handout.4on1.pdf` er begge velegnet til utskrift
- `*.169.pdf`-filene er i 16:9-format
- `*.1610.pdf`-filene er i 16:10-format

- Foredraget er mekket ved hjelp av [GNU Emacs](#), [AUCTEX](#), [pdfTEX](#) fra [MiKTeX](#), [LATEX](#)-dokumentklassa [beamer](#), [Subversion](#), [TortoiseSVN](#) og [Adobe Reader](#)
- Hovedfila bærer denne identifikasjonen:  
`$Ximalas: trunk/zfs-foredrag.tex 10 2014-01-03 12:51:58Z trond $`
- Driverfila for denne PDF-fila bærer denne identifikasjonen:  
`$Ximalas: trunk/zfs-foredrag.foredrag.1610.tex 3 2013-12-23 13:42:53Z trond $`
- Copyright © 2013 Trond Endrestøl
- Dette verket er lisensiert med: [Creative Commons](#), [Navngivelse-DelPåSammeVilkår 3.0 Norge](#) (CC BY-SA 3.0)



# Oversikt over hele foredraget

## Del 1: ZFS?

- 1 Hva er ZFS?
- 2 Et eksempel på en pool
- 3 Hva er grensene til ZFS?
- 4 Hvordan virker ZFS?
- 5 ZFS og RAID-kontrollere
- 6 Hvor kommer ZFS fra?
- 7 Versjonsnummer i ZFS
  - Pool-versjonsnummer
  - Filsystem-versjonsnummer
- 8 Fremtiden for ZFS?

### 9 Administrasjon av ZFS

- zpool
- zfs

### 10 Oppretting av pooler

- Enkle pool-eksempler
- Avanserte pool-eksempler

### 11 zpool-egenskaper

### 12 zfs-egenskaper

Del I

ZFS?

# Oversikt over del 1: ZFS?

- 1 Hva er ZFS?
- 2 Et eksempel på en pool
- 3 Hva er grensene til ZFS?
- 4 Hvordan virker ZFS?
- 5 ZFS og RAID-kontrollere
- 6 Hvor kommer ZFS fra?
- 7 Versjonsnummer i ZFS
  - Pool-versjonsnummer
  - Filsystem-versjonsnummer
- 8 Fremtiden for ZFS?

# Hva er ZFS?



# Hva er ZFS?

- ZFS er

# Hva er ZFS?

- ZFS er
  - 1 Logisk volumhåndterer

(Logical Volume Manager, LVM)

# Hva er ZFS?

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering

# Hva er ZFS?

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer

# Hva er ZFS?

- ZFS er
  - 1 Logisk volumhåndterer (Logical Volume Manager, LVM)
  - 2 Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - 3 Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker

# Hva er ZFS?

- ZFS er
  - 1 Logisk volumhåndterer (Logical Volume Manager, LVM)
  - 2 Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - 3 Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012

# Hva er ZFS?

- ZFS er
  - 1 Logisk volumhåndterer (Logical Volume Manager, LVM)
  - 2 Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - 3 Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av

# Hva er ZFS?

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner



- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disk/partisjoner

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disk/partisjoner
  - ③ Speiling (RAID 1) mellom to eller flere disk/partisjoner

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disker/partisjoner
  - ③ Speiling (RAID 1) mellom to eller flere disker/partisjoner
  - ④ raidz1 (RAID 5, enkel paritet) over tre eller flere disker/partisjoner

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disker/partisjoner
  - ③ Speiling (RAID 1) mellom to eller flere disker/partisjoner
  - ④ raidz1 (RAID 5, enkel paritet) over tre eller flere disker/partisjoner
  - ⑤ raidz2 (RAID 6, dobbel paritet) over fire eller flere disker/partisjoner

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disker/partisjoner
  - ③ Speiling (RAID 1) mellom to eller flere disker/partisjoner
  - ④ raidz1 (RAID 5, enkel paritet) over tre eller flere disker/partisjoner
  - ⑤ raidz2 (RAID 6, dobbel paritet) over fire eller flere disker/partisjoner
  - ⑥ raidz3 («RAID 7», trippel paritet) over fem eller flere disker/partisjoner

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disker/partisjoner
  - ③ Speiling (RAID 1) mellom to eller flere disker/partisjoner
  - ④ raidz1 (RAID 5, enkel paritet) over tre eller flere disker/partisjoner
  - ⑤ raidz2 (RAID 6, dobbel paritet) over fire eller flere disker/partisjoner
  - ⑥ raidz3 («RAID 7», trippel paritet) over fem eller flere disker/partisjoner
- Visse kombinasjoner av det overstående er også mulig

- ZFS er
  - ① Logisk volumhåndterer (Logical Volume Manager, LVM)
  - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
  - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar dataintegritet på alvor, deretter brukervennlighet; hastighet kommer i senere rekker
- Enklere organisering enn «Storage Spaces» i Microsoft Windows Server 2012
- Lagringen organiseres i pooler som kan bestå av
  - ① Enkeltdisker/partisjoner
  - ② Striping (RAID 0) mellom to eller flere disker/partisjoner
  - ③ Speiling (RAID 1) mellom to eller flere disker/partisjoner
  - ④ raidz1 (RAID 5, enkel paritet) over tre eller flere disker/partisjoner
  - ⑤ raidz2 (RAID 6, dobbel paritet) over fire eller flere disker/partisjoner
  - ⑥ raidz3 («RAID 7», trippel paritet) over fem eller flere disker/partisjoner
- Visse kombinasjoner av det overstående er også mulig
- Filsystemet blir opprettet samtidig med poolen

# Et eksempel på en pool



# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

```
errors: No known data errors
```

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata
- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

```
errors: No known data errors
```

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata
- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
- Medlemmene er de tre harddiskene ada2, ada3 og ada4

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata
- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
- Medlemmene er de tre harddiskene ada2, ada3 og ada4
- Poolen har det bare bra og er ONLINE

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

```
errors: No known data errors
```

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
  - Poolen heter enterprise\_zdata
  - Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
  - Medlemmene er de tre harddiskene ada2, ada3 og ada4
  - Poolen har det bare bra og er ONLINE
- Det samme gjelder for vdeven og dens tre medlemmer

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata
- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
- Medlemmene er de tre harddiskene ada2, ada3 og ada4
- Poolen har det bare bra og er ONLINE
- Det samme gjelder for vdeven og dens tre medlemmer
- Null i telleverkene



# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata
- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
- Medlemmene er de tre harddiskene ada2, ada3 og ada4
- Poolen har det bare bra og er ONLINE
- Det samme gjelder for vdeven og dens tre medlemmer
- Null i telleverkene
- Siste skrubbing avsluttet 1. januar 2014, kl. 07:18:15

# Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for status
- Poolen heter enterprise\_zdata
- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
- Medlemmene er de tre harddiskene ada2, ada3 og ada4
- Poolen har det bare bra og er ONLINE
- Det samme gjelder for vdeven og dens tre medlemmer
- Null i telleverkene
- Siste skrubbing avsluttet 1. januar 2014, kl. 07:18:15
- Ingen feil registrert siden 8. januar 2012, kl. 14:14

# Hva er grensene til ZFS?

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt



# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)
  - Maks.  $2^{64}$  enheter tilknyttet en gitt pool

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)
  - Maks.  $2^{64}$  enheter tilknyttet en gitt pool
  - Maks.  $2^{64}$  pooler i et og samme system

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)
  - Maks.  $2^{64}$  enheter tilknyttet en gitt pool
  - Maks.  $2^{64}$  pooler i et og samme system
  - Maks.  $2^{64}$  filsystemer i samme pool

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)
  - Maks.  $2^{64}$  enheter tilknyttet en gitt pool
  - Maks.  $2^{64}$  pooler i et og samme system
  - Maks.  $2^{64}$  filsystemer i samme pool
  - Ref.: <http://en.wikipedia.org/wiki/ZFS>

# Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
  - 128-bit diskadresser
  - Maks.  $2^{48}$  poster i hver katalog
  - Maks.  $2^{64}$  bytes (16 EiB, 16 exbibytes) for hver fil
  - Maks.  $2^{64}$  bytes for hvert attributt
  - Maks.  $2^{78}$  bytes (256 ZiB, 256 zebibytes) i hver pool
  - Maks.  $2^{56}$  attributter for hver fil (egentlig begrenset til  $2^{48}$  attributter)
  - Maks.  $2^{64}$  enheter tilknyttet en gitt pool
  - Maks.  $2^{64}$  pooler i et og samme system
  - Maks.  $2^{64}$  filsystemer i samme pool
  - Ref.: <http://en.wikipedia.org/wiki/ZFS>
- Vis meg det systemet som klarer å sprengre noen av disse grensene!



# Hvordan virker ZFS?

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene
  - 3 Regner ut ny paritet for datablokkene

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene
  - 3 Regner ut ny paritet for datablokkene
  - 4 Skriver oppdatert paritet til de samme paritetsblokkene som tidligere

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strømbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?



# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene
  - 3 Regner ut ny paritet for datablokkene
  - 4 Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strømrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene
  - 3 Regner ut ny paritet for datablokkene
  - 4 Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksummer brukes for alt som blir lagret

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksommer brukes for alt som blir lagret
  - ZFS kontrollerer at leste data er de samme som ble skrevet

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene
  - 3 Regner ut ny paritet for datablokkene
  - 4 Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksommer brukes for alt som blir lagret
  - ZFS kontrollerer at leste data er de samme som ble skrevet
  - Oppdages avvik, leter ZFS etter alternativer

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksommer brukes for alt som blir lagret
  - ZFS kontrollerer at leste data er de samme som ble skrevet
  - Oppdages avvik, leter ZFS etter alternativer
  - Finnes alternativer, enten speilkopier eller paritet, så

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - 1 Skriver nye data til de samme datablokkene som tidligere
  - 2 Leser gamle, urørte data fra de samme datablokkene
  - 3 Regner ut ny paritet for datablokkene
  - 4 Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksommer brukes for alt som blir lagret
  - ZFS kontrollerer at leste data er de samme som ble skrevet
  - Oppdages avvik, leter ZFS etter alternativer
  - Finnes alternativer, enten speilkopier eller paritet, så
    - 1 Leveres korrekte data til applikasjonen, og



- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strømbrydd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksommer brukes for alt som blir lagret
  - ZFS kontrollerer at leste data er de samme som ble skrevet
  - Oppdages avvik, leter ZFS etter alternativer
  - Finnes alternativer, enten speilkopier eller paritet, så
    - ① Leveres korrekte data til applikasjonen, og
    - ② Avviket korrigeres automatisk på den syke disken («resilver»)

# Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
  - ① Skriver nye data til de samme datablokkene som tidligere
  - ② Leser gamle, urørte data fra de samme datablokkene
  - ③ Regner ut ny paritet for datablokkene
  - ④ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
    - Hva skjer *nå* og *senere* hvis du får strømbrydd mellom punktene 1 og 4?
    - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)
- Sjekksommer brukes for alt som blir lagret
  - ZFS kontrollerer at leste data er de samme som ble skrevet
  - Oppdages avvik, leter ZFS etter alternativer
  - Finnes alternativer, enten speilkopier eller paritet, så
    - ① Leveres korrekte data til applikasjonen, og
    - ② Avviket korrigeres automatisk på den syke disken («resilver»)
  - Finnes ingen alternativer, så må filene restaureres fra backup



- Ikke bruk ZFS sammen med RAID-kontrollere!

- Ikke bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS

- Ikke bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
  - RAID-kontrolleren kan finne på å

- **Ikke** bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
  - RAID-kontrolleren kan finne på å
    - Stokke om på skriverekkefølgen

- **Ikke** bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
  - RAID-kontrolleren kan finne på å
    - Stokke om på skriverekkefølgen
    - Utsette skriving av nye data



- Ikke bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
  - RAID-kontrolleren kan finne på å
    - Stokke om på skriverekkefølgen
    - Utsette skriving av nye data
  - Har du skifta batteriet i RAID-kontrolleren?

- Ikke bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
  - RAID-kontrolleren kan finne på å
    - Stokke om på skriverekkefølgen
    - Utsette skriving av nye data
  - Har du skifta batteriet i RAID-kontrolleren?
- Sett RAID-kontrolleren i JBOD-modus, eller

- **Ikke** bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
  - RAID-kontrolleren kan finne på å
    - Stokke om på skriverekkefølgen
    - Utsette skriving av nye data
  - Har du skifta batteriet i RAID-kontrolleren?
- Sett RAID-kontrolleren i JBOD-modus, eller
- La hver harddisk være sitt enslige RAID 0-volum

# Hvor kommer ZFS fra?

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)



# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS
  - ZFS i Linux gjennom FUSE gjenstår som en (treg) mulighet

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS
  - ZFS i Linux gjennom FUSE gjenstår som en (treg) mulighet
  - Brian Behlendorf ved Lawrence Livermore National Laboratory (LLNL) har laget «Native ZFS for/on Linux»

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS
  - ZFS i Linux gjennom FUSE gjenstår som en (treg) mulighet
  - Brian Behlendorf ved Lawrence Livermore National Laboratory (LLNL) har laget «Native ZFS for/on Linux»
- ZFS var tilgjengelig i Mac OS X 10.5, bare read-only, men har vært tilbaketrasket siden oktober 2009

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS
  - ZFS i Linux gjennom FUSE gjenstår som en (treg) mulighet
  - Brian Behlendorf ved Lawrence Livermore National Laboratory (LLNL) har laget «Native ZFS for/on Linux»
- ZFS var tilgjengelig i Mac OS X 10.5, bare read-only, men har vært tilbaketrukket siden oktober 2009
- Noen Mac OS X-entusiaster har laget sine egne ZFS-varianter

# Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS
  - ZFS i Linux gjennom FUSE gjenstår som en (treg) mulighet
  - Brian Behlendorf ved Lawrence Livermore National Laboratory (LLNL) har laget «Native ZFS for/on Linux»
- ZFS var tilgjengelig i Mac OS X 10.5, bare read-only, men har vært tilbaketrasket siden oktober 2009
- Noen Mac OS X-entusiaster har laget sine egne ZFS-varianter
- Andre OS med ZFS-støtte: OpenIndiana, FreeNAS, PC-BSD, GNU/kFreeBSD og NetBSD





- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos

- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos
- Pool-versjonene 29-34 og filsystem-versjon 6 er bare tilgjengelig i Solaris 11 (Express)

- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos
- Pool-versjonene 29-34 og filsystem-versjon 6 er bare tilgjengelig i Solaris 11 (Express)
- OpenSolaris har gått videre til feature-flags og pool-versjon 1000

- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos
- Pool-versjonene 29-34 og filsystem-versjon 6 er bare tilgjengelig i Solaris 11 (Express)
- OpenSolaris har gått videre til feature-flags og pool-versjon 1000
- illumos har gått videre til feature-flags og pool-versjon 5000

- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos
- Pool-versjonene 29-34 og filsystem-versjon 6 er bare tilgjengelig i Solaris 11 (Express)
- OpenSolaris har gått videre til feature-flags og pool-versjon 1000
- illumos har gått videre til feature-flags og pool-versjon 5000
- De fleste OS-er utenom Solaris, samarbeider om videreutviklingen av illumos-varianten

- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos
- Pool-versjonene 29-34 og filsystem-versjon 6 er bare tilgjengelig i Solaris 11 (Express)
- OpenSolaris har gått videre til feature-flags og pool-versjon 1000
- illumos har gått videre til feature-flags og pool-versjon 5000
- De fleste OS-er utenom Solaris, samarbeider om videreutviklingen av illumos-varianten
- Listene på de neste slidene er kopiert fra <http://en.wikipedia.org/wiki/ZFS>

# Versjonsnummer i ZFS I

## Pool-versjonsnummer

- 1 First release
- 2 Ditto Blocks
- 3 Hot spares, double-parity RAID-Z (`raidz2`), improved RAID-Z accounting
- 4 `zpool` history
- 5 `gzip` compression for ZFS datasets
- 6 "`bootfs`" pool property
- 7 ZIL: adds the capability to specify a separate Intent Log device or devices
- 8 ability to delegate `zfs(1M)` administrative tasks to ordinary users
- 9 CIFS server support, dataset quotas
- 10 Devices can be added to a storage pool as "cache devices"
- 11 Improved `zpool scrub/resilver` performance
- 12 Snapshot properties



- 13 Properties: `usedbysnapshots`, `usedbychildren`, `usedbyreservation`, and `usedbydataset`
- 14 `passthrough-x aclinherit` property support
- 15 Properties: `userquota`, `groupquota`, `userused` and `groupused`; also required FS v4
- 16 STMF property support
- 17 triple-parity RAID-Z
- 18 ZFS snapshot holds
- 19 ZFS log device removal
- 20 `zle` compression algorithm that is needed to support the ZFS deduplication properties in ZFS pool version 21, which were released concurrently
- 21 Deduplication
- 22 `zfs receive` properties
- 23 slim ZIL

- 24 System attributes. Symlinks now their own object type. Also requires FS v5.
- 25 Improved pool scrubbing and resilvering statistics
- 26 Improved snapshot deletion performance
- 27 Improved snapshot creation performance (particularly recursive snapshots)
- 28 Multiple virtual device replacements
- 29 RAID-Z/mirror hybrid allocator
- 30 ZFS encryption
- 31 Improved 'zfs list' performance
- 32 One MB block support
- 33 Improved share support
- 34 Sharing with inheritance

- 1 First release
- 2 Enhanced directory entries. In particular, directory entries now store the object type. For example, file, directory, named pipe, and so on, in addition to the object number.
- 3 Support for sharing ZFS file systems over SMB. Case insensitivity support. System attribute support. Integrated anti-virus support.
- 4 Properties: userquota, groupquota, userused and groupused
- 5 System attributes; symlinks now their own object type
- 6 Multilevel file system support

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos



# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD
    - Delphix

# Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD
    - Delphix
    - iXsystems

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD
    - Delphix
    - iXsystems
    - Joyent

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD
    - Delphix
    - iXsystems
    - Joyent
    - NetBSD



- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD
    - Delphix
    - iXsystems
    - Joyent
    - NetBSD
    - Nexenta

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
  - Oracle Solaris
  - illumos/OpenZFS
    - OpenIndiana
    - FreeBSD
    - Delphix
    - iXsystems
    - Joyent
    - NetBSD
    - Nexenta
    - Linux

# Del II

## ZFS!

## 9 Administrasjon av ZFS

- zpool
- zfs

## 10 Oppretting av pooler

- Enkle pool-eksempler
- Avanserte pool-eksempler

## 11 zpool-egenskaper

## 12 zfs-egenskaper



- To kommandoer (med underkommandoer)

- To kommandoer (med underkommandoer)
  - 1 zpool

- To kommandoer (med underkommandoer)
  - ① zpool
    - Administrasjon av lagringspoolene



- To kommandoer (med underkommandoer)
  - 1 zpool
    - Administrasjon av lagringspoolene
  - 2 zfs

- To kommandoer (med underkommandoer)
  - 1 zpool
    - Administrasjon av lagringspoolene
  - 2 zfs
    - Administrasjon av filsystemer, zvolumer, snapshots, kloner, m.m.

- To kommandoer (med underkommandoer)
  - 1 zpool
    - Administrasjon av lagringspoolene
  - 2 zfs
    - Administrasjon av filsystemer, zvolumer, snapshots, kloner, m.m.
- Det finnes en tredje kommando: zdb

- To kommandoer (med underkommandoer)
  - 1 `zpool`
    - Administrasjon av lagringspoolene
  - 2 `zfs`
    - Administrasjon av filsystemer, zvolumer, snapshots, kloner, m.m.
- Det finnes en tredje kommando: `zdb`
  - Brukes for å avlese de indre detaljene til ZFS

- To kommandoer (med underkommandoer)
  - 1 `zpool`
    - Administrasjon av lagringspoolene
  - 2 `zfs`
    - Administrasjon av filsystemer, zvolumer, snapshots, kloner, m.m.
- Det finnes en tredje kommando: `zdb`
  - Brukes for å avlese de indre detaljene til ZFS
  - Bør bare brukes av eksperter ...

- To kommandoer (med underkommandoer)
  - 1 zpool
    - Administrasjon av lagringspoolene
  - 2 zfs
    - Administrasjon av filsystemer, zvolumer, snapshots, kloner, m.m.
- Det finnes en tredje kommando: zdb
  - Brukes for å avlese de indre detaljene til ZFS
  - Bør bare brukes av eksperter ...
  - ... eller av de nysgjerrige

- `zpool add`
  - Brukes for å innføre en helt ny vdev-gruppe med harddisker/partisjoner
- `zpool attach`
  - Brukes for å tilføye en harddisk/partisjon til en eksisterende vdev-gruppe
- `zpool clear`
  - Brukes for å nullstille tellerne for lese-, skrive- og sjekksumfeil
- `zpool create`
  - Brukes for å opprette pooler
- `zpool destroy`
  - Brukes for å ødelegge pooler
- `zpool detach`
  - Brukes for å fjerne en harddisk/partisjon fra en vdev-gruppe
- `zpool export`
  - Brukes for å eksportere en pool, for senere import i samme eller et annet system

- `zpool get`
  - Brukes for å vise verdien til alle eller utvalgte `zpool`-egenskaper
- `zpool history`
  - Brukes for å vise historikken til poolen
- `zpool import`
  - Brukes for å importere en pool eller å vise en liste over pooler som kan importeres
- `zpool iostat`
  - Brukes for å vise I/O-statistikk i sann tid
- `zpool labelclear`
  - Brukes for å fjerne alle spor av ZFS' disklabels
- `zpool list`
  - Brukes for å liste opp importerte pooler
- `zpool offline`
  - Brukes for å deaktivere en harddisk/partisjon
- `zpool online`



- Brukes for (re)aktivere en harddisk/partisjon
- `zpool reguid`
  - Brukes for å tildele en ny, tilfeldig GUID til en bestemt pool
- `zpool remove`
  - Brukes for å fjerne en harddisk/partisjon
- `zpool reopen`
  - Brukes for ...
- `zpool replace`
  - Brukes for å fortelle ZFS at en harddisk/partisjon har blitt skiftet ut
- `zpool scrub`
  - Brukes for å lese gjennom alt aktivt innhold, og sjekke samsvaret mellom lagret data og lagrete sjekksummer
- `zpool set`
  - Brukes for å endre zpool-egenskapene
- `zpool split`

- Brukes for å skille et speilmedlem fra resten av gruppa
- `zpool status`
  - Brukes for å vise status til poolen, dens medlemmer og deres status, og telleverkene for lese-, skrive og sjekksumfeil
- `zpool upgrade`
  - Brukes for å oppgradere poolene til nye formater, vise hvilke pooler som er utdaterte, og hvilke versjoner som er tilgjengelig i systemet

- zfs allow
  -
- zfs bookmark
  -
- zfs clone
  -
- zfs create
  -
- zfs destroy
  -
- zfs diff
  -
- zfs get
  -

- zfs groupspace
  -
- zfs holds
  -
- zfs hold
  -
- zfs inherit
  -
- zfs jail
  -
- zfs list
  -
- zfs mount
  -
- zfs promote

- 
- zfs receive
- 
- zfs release
- 
- zfs rename
- 
- zfs rollback
- 
- zfs send
- 
- zfs set
- 
- zfs share
-

- zfs snapshot
  -
- zfs unallow
  -
- zfs unjail
  -
- zfs unmount
  -
- zfs unshare
  -
- zfs upgrade
  -
- zfs userspace
  -

# Oppretting av pooler

# Oppretting av pooler

- `zpool create [opsjoner] navn-på-pool [organiseringstype] ingredienser [organiseringstype ingredienser] ...`



# Oppretting av pooler

- `zpool create [opsjoner] navn-på-pool [organiseringstype] ingredienser [organiseringstype ingredienser] ...`
- Unngå å plassere mer enn 9 enheter i hver vdev

# Oppretting av pooler

- `zpool create [opsjoner] navn-på-pool [organiseringstype] ingredienser [organiseringstype ingredienser] ...`
- Unngå å plassere mer enn 9 enheter i hver vdev
- I stedet for å stripe en pool over 20 harddisker, vurder å speile to og to harddisker i 10 grupper

# Oppretting av pooler

Enkle pool-eksempler

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:
- `zpool create rpool da0`

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:
- `zpool create rpool da0`
- RAID 0 over to disker:

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:
- `zpool create rpool da0`
- RAID 0 over to disker:
- `zpool create rpool da0 da1`

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:



- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:
  - `zpool create rpool raidz1 da0 da1 da2`

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:
  - `zpool create rpool raidz1 da0 da1 da2`
- RAID 6 over fire disker:

# Oppretting av pooler

## Enkle pool-eksempler

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:
  - `zpool create rpool raidz1 da0 da1 da2`
- RAID 6 over fire disker:
  - `zpool create rpool raidz2 da0 da1 da2 da3`

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:
  - `zpool create rpool raidz1 da0 da1 da2`
- RAID 6 over fire disker:
  - `zpool create rpool raidz2 da0 da1 da2 da3`
- «RAID 7» over fem disker:

- Singledisk:
  - `zpool create rpool da0`
- RAID 0 over to disker:
  - `zpool create rpool da0 da1`
- RAID 1 over to disker:
  - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:
  - `zpool create rpool raidz1 da0 da1 da2`
- RAID 6 over fire disker:
  - `zpool create rpool raidz2 da0 da1 da2 da3`
- «RAID 7» over fem disker:
  - `zpool create rpool raidz3 da0 da1 da2 da3 da4`

# Oppretting av pooler

## Avanserte pool-eksempler



# Oppretting av pooler

## Avanserte pool-eksempler

- RAID ?:

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID ?:
- `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
- `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
- `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID ?:

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
- `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID ?:
- `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
- `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
- `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
- `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
- `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID ?:

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
  - `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
  - `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID ?:
  - `zpool create rpool raidz2 da0 da1 da2 da3 raidz2 da4 da5 da6 da7`



- RAID 1+0 (3 vdevs á 2 disker):
  - `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
  - `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID 6+0 (2 vdevs á 4 disker):
  - `zpool create rpool raidz2 da0 da1 da2 da3 raidz2 da4 da5 da6 da7`

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
  - `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
  - `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID 6+0 (2 vdevs á 4 disker):
  - `zpool create rpool raidz2 da0 da1 da2 da3 raidz2 da4 da5 da6 da7`
- RAID ?:

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
  - `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
  - `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID 6+0 (2 vdevs á 4 disker):
  - `zpool create rpool raidz2 da0 da1 da2 da3 raidz2 da4 da5 da6 da7`
- RAID ?:
  - `zpool create rpool mirror da0 da1 raidz1 da2 da3 da4`

# Oppretting av pooler

## Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
  - `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
  - `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID 6+0 (2 vdevs á 4 disker):
  - `zpool create rpool raidz2 da0 da1 da2 da3 raidz2 da4 da5 da6 da7`
- RAID 1+5+0 (2 vdevs, 2 og 3 disker):
  - `zpool create rpool mirror da0 da1 raidz1 da2 da3 da4`

- size
- capacity
- altroot
- health
- guid
- version
- bootfs
- delegation
- autoreplace
- cachefile
- failmode
- listsnapshots
- autoexpand

- dedupditto
- dedupratio
- free
- allocated
- readonly
- comment
- expandsize
- freeing
- feature@async\_destroy
- feature@empty\_bpobj
- feature@lz4\_compress
- feature@multi\_vdev\_crash\_dump
- feature@spacemap\_histogram
- feature@enabled\_txg

# zpool-egenskaper III

- feature@hole\_birth
- feature@extensible\_dataset
- feature@bookmarks

- type
- creation
- used
- available
- referenced
- compressratio
- mounted
- quota
- reservation
- recordsize
- mountpoint
- sharenfs
- checksum



- compression
- atime
- devices
- exec
- setuid
- readonly
- jailed
- snapdir
- aclmode
- aclinherit
- canmount
- xattr
- copies
- version

- `utf8only`
- `normalization`
- `casesensitivity`
- `vscan`
- `nbmand`
- `sharesmb`
- `refquota`
- `refreservation`
- `primarycache`
- `secondarycache`
- `usedbysnapshots`
- `usedbydataset`
- `usedbychildren`
- `usedbyreservation`

- logbias
- dedup
- mlslabel
- sync
- refcompressratio
- written
- logicalused
- logicalreferenced