

ZFS

Siste ord innen filsystemer

Trond Endrestøl

Fagskolen Innlandet, IT-avdelingen

23. desember 2013

- 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](#), [L^AT_EX](#)-dokumentklassa [beamer](#), [Subversion](#), [TortoiseSVN](#) og [Adobe Reader](#)
- Hovedfila bærer denne identifikasjonen:
`$Ximalas: trunk/zfs-foredrag.tex 6 2013-12-23 19:29:45Z
trond $`
- Driverfila for denne PDF-fila bærer denne identifikasjonen:
`$Ximalas: trunk/zfs-foredrag.handout.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 Hva er grensene til ZFS?
- 3 Hvordan virker ZFS?
- 4 ZFS og RAID-kontrollere
- 5 Hvor kommer ZFS fra?
- 6 Fremtiden for ZFS?

7 Administrasjon av ZFS

8 Oppretting av pooler

- Enkle pool-eksempler
- Avanserte pool-eksempler

Del I

Del 1: ZFS?

Oversikt over del 1: ZFS?

- 1 Hva er ZFS?
- 2 Hva er grensene til ZFS?
- 3 Hvordan virker ZFS?
- 4 ZFS og RAID-kontrollere
- 5 Hvor kommer ZFS fra?
- 6 Fremtiden for ZFS?

Hva er ZFS?

- ZFS er både
 - 1 Logisk volumhåndterer (Logical Volume Manager, LVM)
 - 2 Filsystem med snapshots og kloner
- 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) over tre eller flere disker/partisjoner
 - 5 raidz2 (RAID 6) over seks eller flere disker/partisjoner
 - 6 raidz3 («RAID 7») over ni eller flere disker/partisjoner
- Visse kombinasjoner av det overstående er også mulig

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
- Vis meg det systemet som klarer å sprengre noen av disse grensene!

Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til typiske RAID-kontrollere
 - ① Skriver nye data til de samme datablokkene som tidligere
 - ② Regner ut ny paritet
 - ③ Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
 - Hva skjer hvis du får strømbrydd mellom 1 og 3?
 - Har diskkontrolleren batteribeskyttet minne?
- ZFS skriver fulle striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper
- Sjekksummer 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
 - ① Leveres korrekte data til applikasjonen, og
 - ② avviket korrigeres automatisk på den syke disken
 - Finnes ingen alternativer, så må filene restaureres fra backup

- **Ikke** bruk ZFS sammen med RAID-kontrollere!
- I verste fall kan RAID-kontrolleren motarbeide ZFS
- Sett kontrolleren i JBOD-modus, eller
- la hver disk være sitt enslige RAID 0-volum

Hvor kommer ZFS fra?

- Utviklet av Jeff Bonwick og kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001
- 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
- Andre OS med ZFS-støtte: OpenIndiana, FreeNAS, PC-BSD, GNU/kFreeBSD og NetBSD

Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, 27. januar 2010
- Oracle ville gjøre OpenSolaris om til «ClosedSolaris»
- Hele ZFS-teamet hos Oracle sa opp på dagen, 90 dager etter den avgjørelsen
- ZFS lever videre hos
 - Oracle
 - illumos
 - OpenZFS
 - FreeBSD
 - Delphix
 - iXsystems
 - Joyent
 - NetBSD
 - Nexenta

Del II

Del 2: ZFS!

7 Administrasjon av ZFS

8 Oppretting av pooler

- Enkle pool-eksempler
- Avanserte pool-eksempler

- To kommandoer (med underkommandoer):
 - 1 `zpool`
 - 2 `zfs`
- Det finnes en tredje kommando for de nysgjerrige: `zdb`
 - Brukes for å avlese indre ZFS-detalljer

- `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

- 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 seks disker:
 - `zpool create rpool raidz2 da0 da1 da2 da3 da4 da5`
- «RAID 7» over ni disker:
 - `zpool create rpool raidz3 da0 da1 da2 da3 da4 da5 da6 da7 da8`

Oppretting av pooler

Avanserte pool-eksempler

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