

ZFS met RAID-Z als alternatief voor klassieke RAID-oplossingen

Onderzoeksvoorstel Bachelorproef

Jonas De Moor¹

Samenvatting

Bij bedrijven worden RAID-oplossingen al jarenlang gebruikt voor bescherming tegen dataverlies. Klassieke RAID kent echter een aantal inherente problemen, waaronder risico op corruptie van data bij een systeemcrash. In 2002 ontwikkelde het toenmalige Sun Microsystems ZFS, een nieuw bestandssysteem voor Solaris dat lang bestaande problemen met bestandssystemen van de laatste 20 jaar moet oplossen. ZFS biedt naast een ingebouwde volume manager en storage pools nog een hoop andere functionaliteiten, waaronder RAID-Z, een softwarematige RAID-oplossing die niet gebukt gaat onder de problemen van klassieke RAID-oplossingen. Ondertussen heeft een groep ontwikkelaars zich opgesplitst van Oracle en werd het OpenZFS project opgericht met de bedoeling om ZFS te ontwikkelen op een open manier. Sinds 2013 is ZFS ook beschikbaar voor Linux, en in 2016 begon Canonical van de populaire distributie Ubuntu met plannen om ZFS op te nemen in de distributie. Dit onderzoek zal de architectuur van ZFS bespreken, samen met de belangrijkste verschillen tussen ZFS, RAID en "klassieke" bestandssystemen. Ook wordt er een praktisch onderzoek verricht om zo de performantie en betrouwbaarheid van ZFS in kaart te brengen.

Sleutelwoorden

Systeem- en Netwerkbeheer — Opslagapparaten — Serverhardware — Bestandssystemen — ZFS — Unix

Contact: ¹ jonas.demoor.v3741@student.hogent.be

Inhoudsopgave

1	Introductie	1
2	State-of-the-art	2
2.1	The Zettabyte Filesystem (e. a. Jeff Bonwick, 2002)	2
2.2	ZFS and RAID-Z: The Über-FS? (Brian Hickmann, 2007)	2
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2
	Referenties	2

1. Introductie

Al jarenlang is de meest gebruikte oplossing tegen dataverlies door het falen van schijven RAID, wat staat voor Redundant Array of Independent (of Inexpensive) Disks. RAID is echter niet geheel feilloos: het beschermt bv. niet tegen fouten die gemaakt zijn door gebruikers (denk maar aan het wissen van belangrijke data) en het biedt ook geen oplossing als er in een zelfde tijdsbestek meerdere schijven falen (Peter M. Chen, 1993).

In 2002 begon het toenmalige Sun Microsystems, nu onderdeel van Oracle Corporation, aan de ontwikkeling van ZFS

(Zettabyte Filesystem). Dit is een bestandssysteem dat volledig *from scratch* is ontwikkeld om “de tekortkomingen van huidige bestandssystemen op te lossen” (B. M. Jeff Bonwick, Unknown), vooral met betrekking tot data-integriteit. Volgens de toenmalige hoofdontwikkelaar e. a. Jeff Bonwick (2002) biedt ZFS heel wat vernieuwende functionaliteiten zoals een eenvoudiger beheer, automatische foutcorrectie, automatisch schalende partities en een softwarematige RAID onder de term *RAID-Z*.

Sinds 2013 is ZFS ook beschikbaar voor Linux, maar de eerste uitgaven leden onder nogal wat stabiliteitsproblemen. Tegenwoordig is ZFS op Linux volwassen genoeg geworden om in te zetten in productie. Deze bachelorproef is dan ook een goede gelegenheid om zelf wat meer onderzoek te doen naar ZFS en of het een volwaardig alternatief zou zijn voor een klassieke, hardwaregebaseerde RAID-oplossing op een server.

Dit onderzoek zal trachten een antwoord te vinden op volgende vragen:

- Wat zijn de grootste verschillen tussen een klassieke RAID-oplossing en ZFS RAID-Z? Hoe is de architectuur van ZFS opgebouwd en op welke manieren tracht het oplossingen te vinden voor de problemen die zich voordoen bij andere bestandssystemen en RAID-opstellingen?

- Komt ZFS zijn beloften qua data-integriteit en prestatie na onder verschillende workloads en toepassingen?

2. State-of-the-art

2.1 *The Zettabyte Filesystem* (e. a. Jeff Bonwick, 2002)

Dit is een paper geschreven door de oorspronkelijke ontwikkelaars van ZFS. In deze paper wordt er een gedetailleerd overzicht gegeven van de architectuur van het bestandssysteem en worden klassieke problemen met andere bestandssystemen aangekaart. Er worden voortdurend vergelijkingen gemaakt tussen de manier waarop ZFS problemen aanpakt en de manier waarop andere bestandssystemen dit probleem aanpakken. Zo is het volgens e. a. Jeff Bonwick (2002) bv. niet acceptabel om bij journaalgebaseerde bestandssystemen (zoals EXT4) enkel te vertrouwen op logs en het draaien van `fsck` als er eventuele corruptie van data zou optreden. Dit omdat volgens deze redenering dan alle recovery-code in de bootloader zich zou moeten bevinden, want het is perfect mogelijk dat het bestandssysteem met het besturingssysteem corrupt kan geraken.

Als conclusie stellen de auteurs dat ZFS heel wat problemen van de meer “klassieke” bestandssystemen (zoals moeilijk beheer en slechte data integriteit) uit de weg gaat d.m.v. een aantal architecturale beslissingen (zoals een object-gebaseerd model, storage pools en uitgebreide checksumming op block-niveau) (e. a. Jeff Bonwick, 2002).

2.2 *ZFS and RAID-Z: The Über-FS?* (Brian Hickmann, 2007)

Dit is een paper geschreven door studenten aan de universiteit van Wisconsin-Madison en is geschreven gedurende de periode dat ZFS nog maar net was uitgebracht. Er waren nog niet zoveel details gepubliceerd over RAID-Z door Sun, dus besloten deze studenten om zelf onderzoek te doen naar de interne werking van deze ZFS functionaliteit. Ook analyseren ze de prestatie van een ZFS RAID bij hoge fragmentatie en bij bijna volle schijven. Hierbij kwamen ze tot de conclusie dat prestatie heel wat slechter is dan in andere gevallen. Daarnaast onderzochten de studenten of ZFS wel degelijk een oplossing biedt voor het RAID 5 “write hole” probleem, waarbij data in een inconsistente staat kan geraken als het systeem bijvoorbeeld crasht. Ze concludeerden dat RAID-Z wel degelijk een oplossing bood voor dit probleem.

Deze paper komt in grote lijnen overeen met dingen die ik ook zou willen testen. Maar waar deze paper niet over spreekt is prestatie bij verschillende toepassingen (zoals databases en virtualisatie); dit is een aspect dat ik graag zou onderzoeken.

3. Methodologie

Naast een theoretisch deel dat vooral zal handelen over de architectuur van ZFS en de werking van zowel hardwarematige als softwarematige RAID, zal er ook een praktisch

deel zijn. In dit deel zal ik de prestatie (vooral disk I/O prestatie) van RAID-Z vergelijken met die van een hardwarematige RAID 5. Standaard RAID-Z komt overeen met RAID 5 qua functionaliteit. Daarnaast ga ik ook de prestatie van verschillende toepassingen onder ZFS vergelijken met de prestatie van deze toepassingen onder “gewone” bestandssystemen (zoals EXT4 en XFS). Bij deze laatstgenoemde vergelijking zal er geen RAID-opstelling worden gebruikt en zal ZFS dus op slechts één schijf draaien.

Voor deze testen ga ik gebruikmaken van verschillende tools (zoals `dd`) en benchmarks om de prestatie te meten. Prestatie zal in dit geval dus vooral I/O-performance en response time bij verschillende toepassingen zijn. Bij dit onderzoek ga ik gebruikmaken van een fysieke server met een RAID-controller die beschikbaar zal worden gesteld door DVIT of een werknemer van DVIT.

4. Verwachte resultaten

Ik verwacht dat hardwaregebaseerde RAID iets betere prestatie zal hebben dan RAID-Z. Maar qua betrouwbaarheid zal RAID-Z dan weer een streepje voor hebben. Bij het testen van verschillende soorten toepassingen op ZFS verwacht ik dat de resultaten niet al teveel zullen verschillen van de resultaten bij het draaien van deze toepassingen op “gewone” bestandssystemen. Enkel bij databases zou dit wel eens een grotere impact kunnen hebben.

5. Verwachte conclusies

Ik verwacht dat ik ZFS met RAID-Z als volwaardig (en misschien zelfs goedkoper) alternatief voor klassieke RAID ga kunnen bestempelen. Ook al zou ZFS misschien een goed alternatief (of zelfs beter) zijn dan hardwaregebaseerde RAID, dan betekent dit niet dat “gewone” RAID geen plaats meer zou hebben. Een argument tegen ZFS zou bijvoorbeeld kunnen zijn dat het bestandssysteem enkel beschikbaar is voor Unix en Linux. Ik verwacht dat RAID dus nog voor lange tijd zijn plaats zal hebben in de opslagwereld.

Referenties

- Brian Hickmann, K. S. (2007). *ZFS and RAID-Z: The Über-FS?* Verkregen van <http://pages.cs.wisc.edu/~remzi/Classes/736/Fall2007/Projects/BrianKynan/paper.pdf>
- Jeff Bonwick, B. M. (Unknown). *ZFS The Last Word In File Systems*. Verkregen van <https://wiki.illumos.org/download/attachments/1146951/zfs.last.pdf>
- Jeff Bonwick, e. a. (2002). *The Zettabyte Filesystem*. Verkregen van <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.184.3704&rep=rep1&type=pdf>
- Peter M. Chen, e. a. (1993). *RAID: High-Performance, Reliable Secondary Storage*. Verkregen van <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.3889&rep=rep1&type=pdf>