# Louwrentius



## ZFS RAIDZ Expansion Is Awesome but Has a Small Caveat

Tue 22 June 2021 Category: ZFS

(2021-06-22T12:00:00+02:00)

## Introduction

One of my most popular blog articles is this article (https://louwrentius.com/the-hidden-cost-of-using-zfs-for-your-home-nas.html) about the "Hidden Cost of using ZFS for your home NAS". To summarise the key argument of this article:

*Expanding ZFS-based storge can be relatively expensive / inefficient.*

For example, if you run a ZFS pool based on a single 3-disk RAIDZ vdev (RAID5 equivalent[1] (#fn:raidz)), the only way to expand a pool is to add another 3-disk RAIDZ vdev.

You can't just add a single disk to the existing 3-disk RAIDZ vdev to create a 4-disk RAIDZ vdev because vdevs can't be expanded.

The impact of this limitation is that you have to buy all storage upfront even if you don't need the space for years to come.

Otherwise, by expanding with additional vdevs you lose capacity to parity you may not really want/need, which also limits the maximum usable capacity of your NAS.

## RAIDZ vdev expansion

Fortunately, this limitation of ZFS is being addressed!

ZFS founder Matthew Ahrens created a pull request (https://github.com/openzfs/zfs/pull/12225) around June 11, 2021 detailing a new ZFS feature that would allow for RAIDZ vdev expansion.

Finally, ZFS users will be able to expand their storage by adding just one single drive at a time. This feature will make it possible to expand storage as-you-go, which is especially of interest to budget conscious home users[2 (#fn:larger)].

Jim Salter (https://arstechnica.com/author/jimsalter/) has written a good article (https://arstechnica.com/gadgets/2021/06/raidz-expansion-code-lands-in-openzfs-master/) about this on Ars Technica.

## There is still a caveat

Existing data will be *redistributed* or *rebalanced* over all drives, including the freshly added drive. However, the data that was already stored on the vdev will not be *restriped* after the vdev is expanded. This means that this data is stored with the older, *less efficient* parity-to-data ratio.

I think Matthew Ahrends explains it best in his own words:

**Solar Status**

| | |
|---|---|
| Status | Charging |
| Solar | 15 W |
| Load | 3.12 W |
| Load Current | 238 mA |
| Battery | 97 % |
| Bat. Current | 882 mA |
| Bat. Voltage | 13.5 V |
| Uptime | 31 days |
| Remaining | 53 H |

**71 TiB NAS**

**20C/40T 128G Server**

**Projects**

- fio-plot
- Showtools
- Storage Fan Control
- Grafana Dasboard for storage metrics

**Categories**

- Apple
- Bash
- Blogging
- Debian
- FreeBSD
- Hardware
- Iptables
- Linux
- Lustre
- Microsoft
- Monitoring

```
After the expansion completes, old blocks remain with their old data-to-parity ratio (e.g. 5
-wide RAIDZ2, has 3 data to 2 parity), but distributed among the larger set of disks. New bl
ocks will be written with the new data-to-parity ratio (e.g. a 5-wide RAIDZ2 which has been
expanded once to 6-wide, has 4 data to 2 parity). However, the RAIDZ vdev's "assumed parity
ratio" does not change, so slightly less space than is expected may be reported for newly-wr
itten blocks, according to zfs list, df, ls -s, and similar tools.
```

So, if you add a new drive to a RAIDZ vdev, you'll notice that after expansion, you will have *less* capacity available than you would theoretically expect.

However, it is even more important to understand that this effect *accumulates*. This is especially relevant for home users.

I think that the whole concept of starting with a small number of disks and expand-as-you-go is very desirable and typical for home users. But this also means that every time a disk is added to the vdev, existing data is still stored with the old data-to-parity rate.

*Imagine that we have a 10-drive chassis and we start out with a 4-drive RAIDZ2.*

If we keep adding drives[4 (#fn:gofrom)] conform this example, until the chassis is full at 10 drives, about 1.35 drives worth of capacity is 'lost' to parity overhead/efficiency loss[3 (#fn:full)].

That is quite a lot of overhead or loss of capacity, I think.

How is this overhead calculated? If we would just buy 10 drives and create a 10-drive RAIDZ2 vdev, data-to-parity overhead is 20% meaning that 20% of the total raw capacity of the vdev is used for storing parity. This is the most efficient scenario in this case.

When we start out with the four-drive RAIDZ2 vdev, the data-to-parity overhead is 50%. That's a 30% overhead difference compared to the 'ideal' 10-drive setup.

As we keep adding drives, the relative overhead of the parity keeps dropping so we end up with 'multiple data sets' with different data-to-parity ratios, that are less efficient than the end-stage of 10 drives.

I created a google sheet to roughly estimate this overhead for each stage, but my math was totally off. Fortunately, Yorick (https://www.truenas.com/community/members/yorick.90628/) rewrote the sheet, which can be found here (https://docs.google.com/spreadsheets/d/1qiDPfLN-K88FMHMxcgtkxswY5Wtu7h9tBAOgJfnO7VE/edit?usp=sharing). Thanks Yorick! Further more, Truenas user DayBlur shared additional insights (https://www.truenas.com/community/threads/raidz-expansion-its-happening.58575/post-649578) on the calculations if you are interested in that.

The google sheet (https://docs.google.com/spreadsheets/d/1qiDPfLN-K88FMHMxcgtkxswY5Wtu7h9tBAOgJfnO7VE/edit?usp=sharing) allows you to play with various variables to estimate how much capacity is lost for a given scenario. Please note that any losses that may arise because a number of drives is used that requires data to be padded - as discussed in the Ars Technica article - are not part of the calculation.

It is a bit unfortunate that especially in the scenario of the home user who want to start small and expand-as-you go that this overhead manifests itself so much. But there is good news!

## Lost capacity can be recovered!

The overhead or 'lost capacity' can be *recovered* by *rewriting* existing data after the vdev has been expanded, because the data will then be written with the more efficient parity-to-data ratio of the larger vdev.

Rewriting all data may take quite some time and you may opt to postpone this step until the vdev has been expanded a couple of times so the parity-to-data ratio is now 'good enough' that significant storage gains can be had by rewriting the data.


- NAS
- Networking
- OpenVPN
- PPSS
- Proxy
- RAID
- Robotics
- security
- Server
- Smartphone
- Software
- solar
- Solaris
- Storage
- Technology
- Uncategorized
- VMware
- Windows
- ZFS

**Archive**

- 2021
- 2020
- 2019
- 2018
- 2017
- 2016
- 2015
- 2014
- 2013
- 2012
- 2011
- 2010
- 2009
- 2008

**Social**

- louwrentius

Because capacity lost to overhead can be fully recovered, I think that this caveat is relatively minor, especially compared to the old situation where we had to expand a pool with entire vdevs and there was no way to recover any overhead.

There is currently no build-in mechanism to trigger this data rewrite as part of the native ZFS tools. This will be a manual process until somebody may create a script that automates this process. According to Matthew Ahrens, restriping the data would be an effort of similar scale (https://github.com/openzfs/zfs/pull/12225#issuecomment-860075460) as the RAIDZ expansion itself.

## Evaluation

I think it cannot be stated enough how *awesome* the RAIDZ vdev expansion feature is, especially for home users who want to start small and grow their storage over time.

Although the expansion process can accumulate quite a bit of overhead, that overhead can be recovered by rewriting existing data, which is probably not a problem for most people.

Despite all the awesome features and capabilities of ZFS, I think quite a few home users went with other storage solutions because of the relatively high expansion cost/overhead. Now that this barrier will be overcome, I think that ZFS will be more accessible to the home user DIY NAS crowd.

## Release timeline

According to the Ars Technica article by Jim Salter, this feature will probably become available in August 2022, so we need to have some patience. Even so, you might want to already decide to build your new DIY NAS based on ZFS: by the time you may need to expand your storage, the feature may be available!

---

1. Just to illustrate the level of redundancy in terms of how many disks can be lost and still be operational. ↩ (#fnref:raidz)

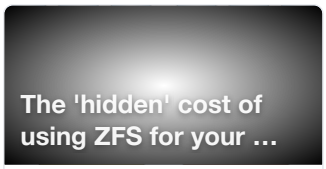
2. I personally think that it's even great for small and medium business owners. Only larger businesses want to keep adding relatively large vdevs consisting of multiple drives because if they keep expanding with just one drive at a time, they may have to expand capacity very frequently which may not be practical. ↩ (#fnref:larger)

3. If you would only upgrade once the pool is almost full - not recommended! - that overhead grows to 1.69 drives. ↩ (#fnref:full)

4. So you go from four to five drives. Then from five to six drives, and so on. ↩ (#fnref:gofrom)

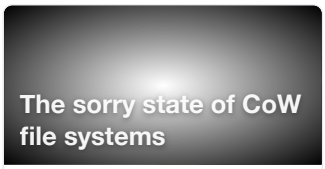
### Comments

ALSO ON **LOUWRENTIUS**



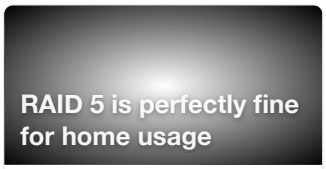

5 years ago • 119 comments

Update October 2017 | Please note that RAIDZ expansion is under …



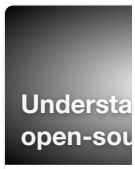

6 years ago • 46 comments

I'd like to argue that both ZFS and BTRFS both are incomplete file systems …



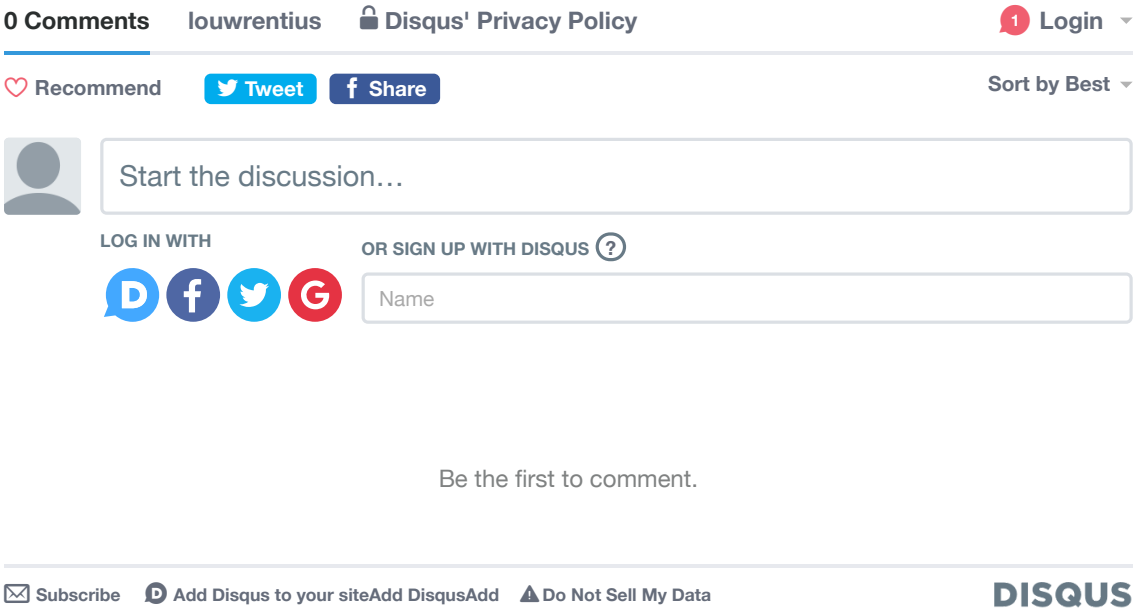

5 years ago • 14 comments

RAID 5 gets a lot of flak these days. You either run RAID 1, RAID 10 or you …




3 years ago

Introductio
post I will t
I believe Ce




Proudly powered by Pelican, which takes great advantage of Python.
Based on the Gumby Framework