Test for read/write speeds, array rebuild times, scrub times on 2-6 hard drives for various array types

How do the ZFS pool configurations compare to each other based on a number of disks

Papers read:

28/11/2023

**ZFS STORAGE POOL LAYOUT**

<https://static.ixsystems.co/uploads/2020/09/ZFS_Storage_Pool_Layout_White_Paper_2020_WEB.pdf>

Positives:

* Clearly explains the different types of pool configurations
* Shows formulas and examples for working out read/write speeds, iops, redundancy levels and capacity

Negatives:

* Only gives theoretical numbers, no real world figures, arguably unimportant and performance figures will depend on a variety of factors, although things like CPU are unlikely to affect things these days
* Does not include resilvering times and probability of loosing the pool

How it relates

* Basically what I want to do plus writing a script to give some real world performance metrics

21/11/2023

**A PERFORMANCE COMPARISON**

**OF ZFS AND BTRFS ON LINUX**

<https://www.diva-portal.org/smash/get/diva2:822493/FULLTEXT01.pdf>

Positves:

* Points out using warm cache
* Experiments done 20 times, discarding first time to reach steady-state
* Important to think about cache and memory

Negatives:

* Results consistently paint ZFS as the worst which is not comparable to many other results
* From 2015, more up to date than other papers but still out of date, would be interesting to see if the performance numbers match more recent results

**The Zettabyte Filesystem**

<https://www.cs.hmc.edu/~rhodes/cs134/readings/The%20Zettabyte%20File%20System.pdf>

Positives:

* A paper about ZFS from the creators highlighting what makes ZFS different and how it works

Negatives:

* Biased

**Benchmarking file system benchmarking: it \*IS\* rocket science**

24/11/2023

<https://www.researchgate.net/publication/262350224_Benchmarking_file_system_benchmarking_it_IS_rocket_science>

How it relates:

* Seems to show IOmeter would be best used to measure IOPS
* Can talk about how it points out that some benchmarks like Filebench are unable to separate between different levels of the file system, however it is what most others use so the results are more valuable when able to compare them with others
* Is it necessary to separate the parts out
* All parts of the filesystem are so intertwined that it is difficult to separate and test the individual parts separately
* Makes interesting points about caching, need to limit the amount of cache otherwise the files can fit entirely in memory, at which point the speeds comes from the memory system not the file system
* Interesting to test how the file system uses cache but hard to control

Existing benchmarks

**On the implementation of ZFS (Zettabyte File System) storage system**

25/11/2023

<https://www.researchgate.net/publication/315866344_On_the_implementation_of_ZFS_Zettabyte_File_System_storage_system>

Positives:

* Provides very granular results between different ZFS compression algorithms

Negatives:

* According to the benchmarking rocket science paper, bonnie++ is not able to isolate the on-disk and caching layers, i.e. the performance will improve as time goes on as files begin to be cached
* According to the open source github repository, LZ4 has had performance improvements since this papers publication <https://github.com/lz4/lz4/releases>

How it relates:

* Shows that, with the exception of very specific circumstances such as very low available CPU resources or entirely writing compressible files, LZ4 is almost always the best compression algorithm to use and so would not be worth repeating the tests with different compression algorithms

## \*\*\*[ZFS: Resilver Performance of Various RAID Schemas](https://louwrentius.com/zfs-resilver-performance-of-various-raid-schemas.html)

26/11/2023

<https://louwrentius.com/zfs-resilver-performance-of-various-raid-schemas.html>

Positive:

* Tests a wide variety of pool configurations and number of disks to find resilvering times
* Tests resilvering times at both 25% and 50%

Negatives:

* Does not test anything more than 50%
* Only tests data immediately after it has been written to disk and where it is not fragmented. In reality, by the time a disk failed and the array would need rebuilding, the data would be heavily fragmented which could lead to increased resilvering times
* Is not stated whether the test has been run multiple times to validate results
* RAID-Z tests 3, 4, 5 and 9 disks, RAID-Z2 tests 4, 6 and 10 and RAID-Z3 tests 5, 7 and 11. Would have been nice to see some consistency with the amount of disks being used

How it relates to my question:

* Provides some values to compare my results against
* My results will be more granular as I will test every possible configuration on 2-6 drives
* Will likely just fill the array to 10% as this research shows that as the array fills up, resilvering times mostly increase linearly with some extra as the disk tracks become closer to the center

**\*\*\*Workload Dependent Performance Evaluation of the Btrfs and ZFS Filesystems**

<https://picture.iczhiku.com/resource/paper/shIdkpUhfRUjHbcx.pdf>

Positives:

* Provides some relevant benchmarks
* Explains some of the architecture for ZFS
* Repeated benchmarks 10 times

Negatives:

* Published in 2009 so rather out of date

How it relates:

* Provides some relevant benchmarks to compare to
* Can talk about how it gave me some background on ZFS
* Mentions that zfs shouldn’t be used for situations where there aren’t a lot of cpu resources

**Performance Evaluation of ZFS and LVM (with ext4)**

**for Scalable Storage System**

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5930130>

positives:

* Provides some benchmarks
* Possible the ZFS over sata results are relevant to me
* Demonstrates that ZFS gives better performance the more drives it has due to its dynamic striping, relevant to my research as I can talk about this and compare it to whether ZFS does give better “throughput”

Negatives:

* Only benchmarks 1-3 drives
* Emphasis is on comparing LVM to ZFS over the network so may not be completely relevant

**Ext4, XFS, BtrFS and ZFS Linux File Systems on**

**RADOS Block Devices (RBD):**

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7881982>

Positives:

* Gives benchmarks
* Uses tools that others have used such as filebench so results can be comparable

Negaives:

* Paper focuses on using RADOS so may not strictly be relevant
* They apparently use RAID 0 for ZFS which is an odd choiuce

A Review on Performance Analysis of ZFS & BTRFS

[https://ieeexplore.ieee.org/---------------------------------------stamp/stamp.jsp?tp=&arnumber=8698103](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8698103)

more of a literature review, does not do any kind of analysis or experiment, just sums up other papers

Could test the performance of ZFS on one drive, is a little redundant though

2 Disks, Mirror, Read/Write, IOPS, resilver speed

2 Disks RAID-Z, Read/Write, IOPS, resilver speed

3 Disks, Mirror, Read/Write, IOPS, resilver speed

3 Disks, RAID-Z, Read/Write, IOPS, resilver speed

3 Disks, RAID-Z2, Read/Write, IOPS, resilver speed

4 Disks, Mirror, Read/Write, IOPS, resilver speed

4 Disks, RAID-Z, Read/Write, IOPS, resilver speed

4 Disks, RAID-Z2, Read/Write, IOPS, resilver speed

4 Disks, RAID-Z3, Read/Write, IOPS, resilver speed

5 Disks, Mirror, Read/Write, IOPS, resilver speed

5 Disks, RAID-Z, Read/Write, IOPS, resilver speed

5 Disks, RAID-Z2, Read/Write, IOPS, resilver speed

5 Disks, RAID-Z3, Read/Write, IOPS, resilver speed

6 Disks, Mirror, Read/Write, IOPS, resilver speed

6 Disks, RAID-Z, Read/Write, IOPS, resilver speed

6 Disks, RAID-Z2, Read/Write, IOPS, resilver speed

6 Disks, RAID-Z3, Read/Write, IOPS, resilver speed

Is it worth testing stripe? Would never be used in the real world as if one disk dies, all data is lost

What about a striped mirror?

Could do some benchmarks limiting memory and without

A black screen with a white line

Description automatically generated