napp-*it*

Allocation Classes Feature

Benchmarks and use case on OmniOS

published: 2019, Oct 31 (c) napp-it.org

Allocation Classes

Content:

1. About Allocation Classes // Performance impact

Allocation classes is a Open-ZFS feature initiated by Intel to isolate large block file data on a regular datapool from metadata, small io transfers and dedup tables by using different types of vdevs for different types of data. This is an alternative approach to data tiering where you move a whole file that requires a better performance to a faster part of an array.

Allocation Classes improve performance not for a whole file and with a needed file move like Tiering but improve performance based on type of data or recordsize of a filesystem. Performance sensitive data like metadata, small io or dedup tables can be placed on an ultrafast vdev while more uncritical large data remain on the quite slower regular vdevs like mirrors or Raid-Z.

more: https://zfs.datto.com/2017_slides/brady.pdf

Allocation Classes	**Purpose**
Normal (Basic, mirror, Raid-Z)	Any block type
Log	ZIL records
Metadata Allocation (new)	Pool/ Filesystem metadata
Dedup Table (new)	Deduplication Table Data (DDT)
Small Blocks	Small block sizes (0,1K-128K)

L2Arc is an extension for the rambased readcache Arc for small random reads while Allocation Classes store data based on datatype.

Allocation classes where small random io is placed to a high performance vdev can boost performance on slow disks pools at a fraction of the price of full SSD pools. The key setting is the ZFS dataset property „special_small_blocks"=size (512B up to 1M.
The default size is 0 which means no small file blocks will be allocated in the special class. This will improve only access to metadata. If you set special_small_blocks" then all data with a recordsize smaller or equal this setting will land on the special vdev.

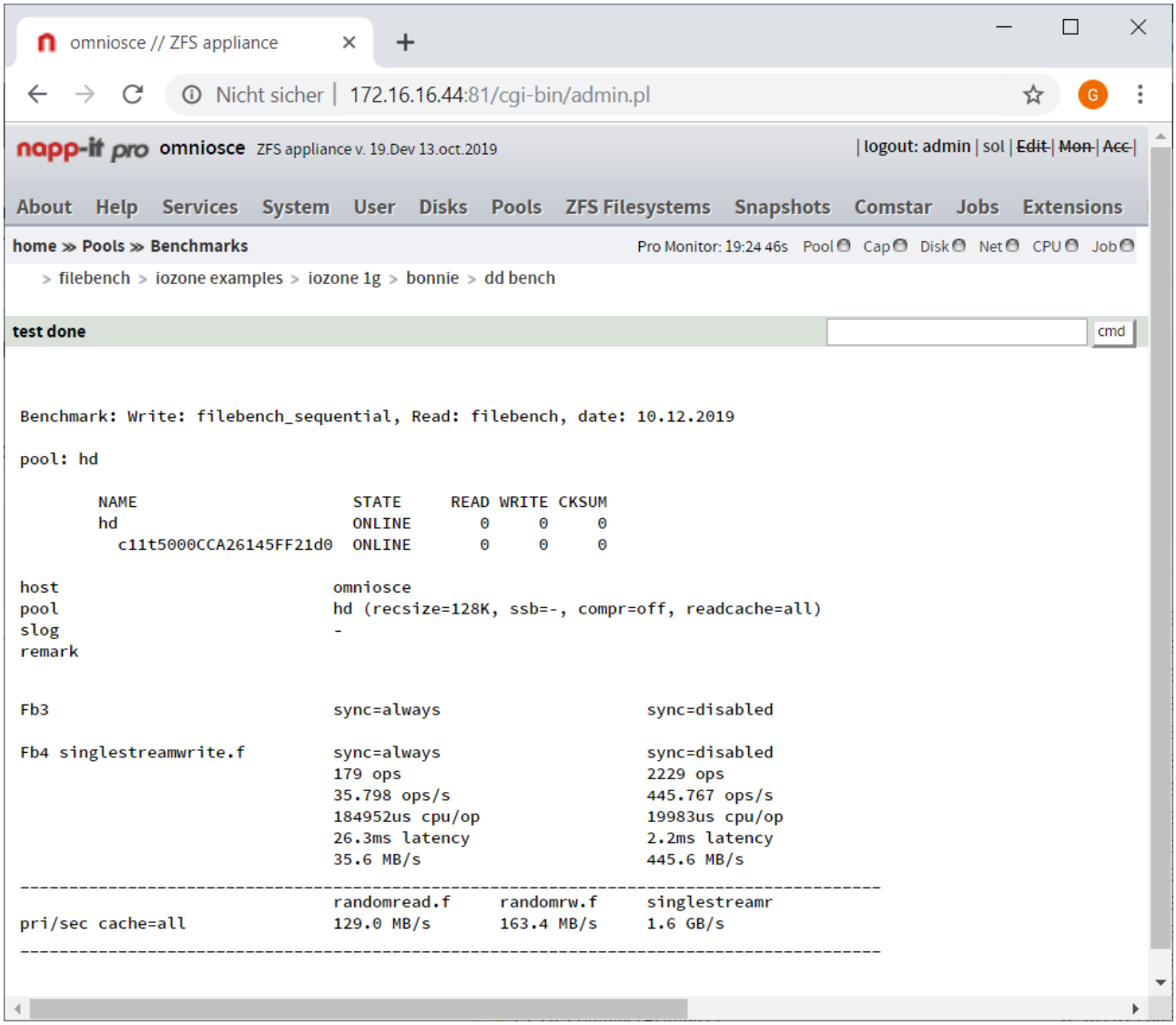
If you enable compress, blocksize can be variable/smaller. Effect of this needs some more testings. Effectively there may be more data then on the special vdev.

As special vdevs are there to store part of pooldata, their redundancy level must be equal to other pool vdevs. Usually this means that special vdevs should be n-way mirrors. If a special vdev gets full, ZFS will automatically use the regular vdevs. If you use more than one special vdev, load is balanced over them.

Special and dedup vdevs can be removed (Mirror and basic) but only if all vdevs have the same ashift. To be sure, force all vdevs to same ashift ex ashift=12 (4k disks).

In my tests with OmniOS bloody, the OS crashed and the pool was damaged when I tried to remove an ashift=9 special vdev from a pool with ashift=12 vdevs.

2. Diskbased pool without Slog or special vdev (basic vdev)

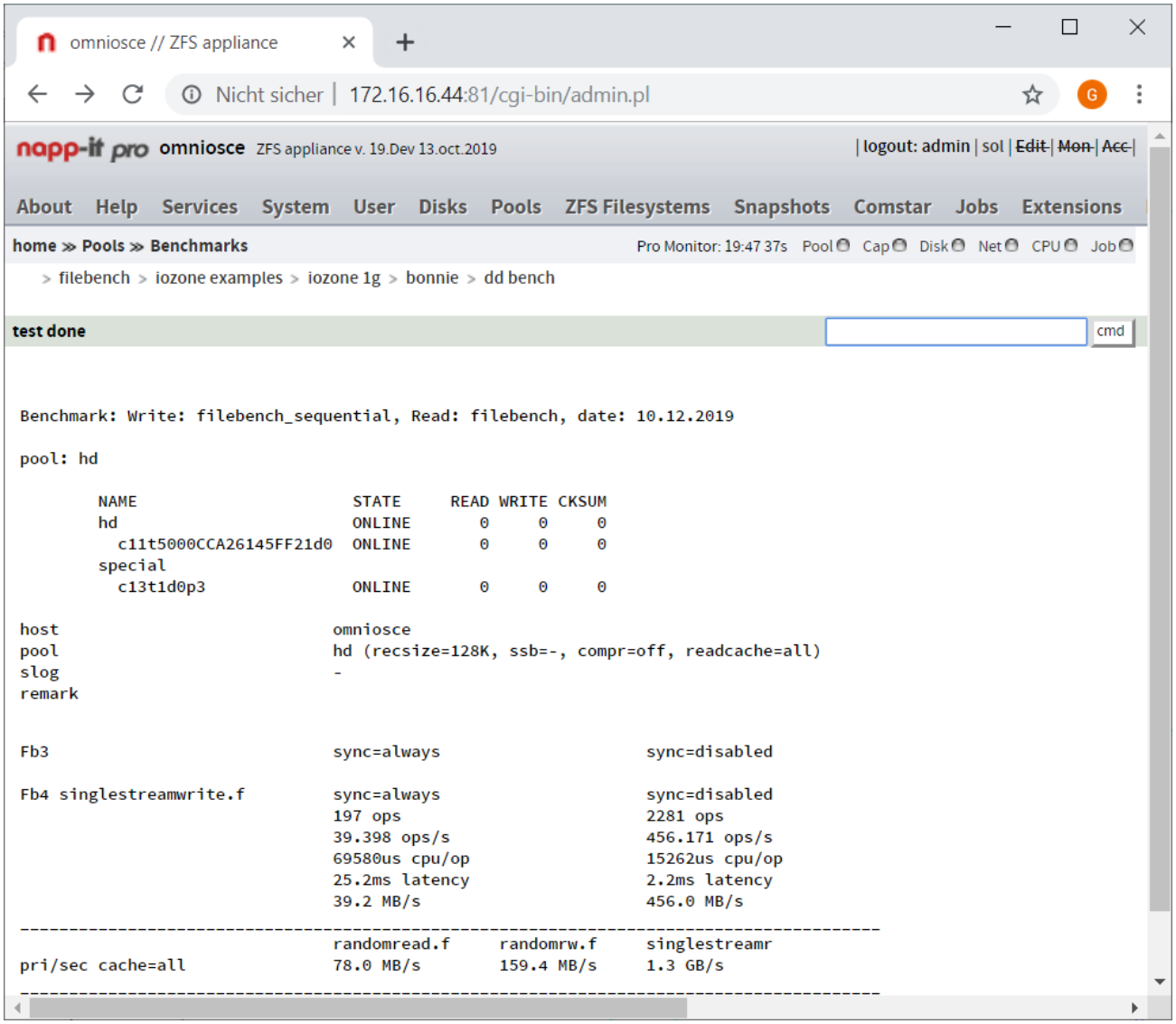


The test environment is ESXi 6.7U3 with a OmniOS bloody VM 151031 (October) with a pool from a single HGST HE8 disk on an LSI 2008 HBA in pass-through mode. I assigned 4 cores and 16 GB RAM to the VM.

Sync write is a mess.

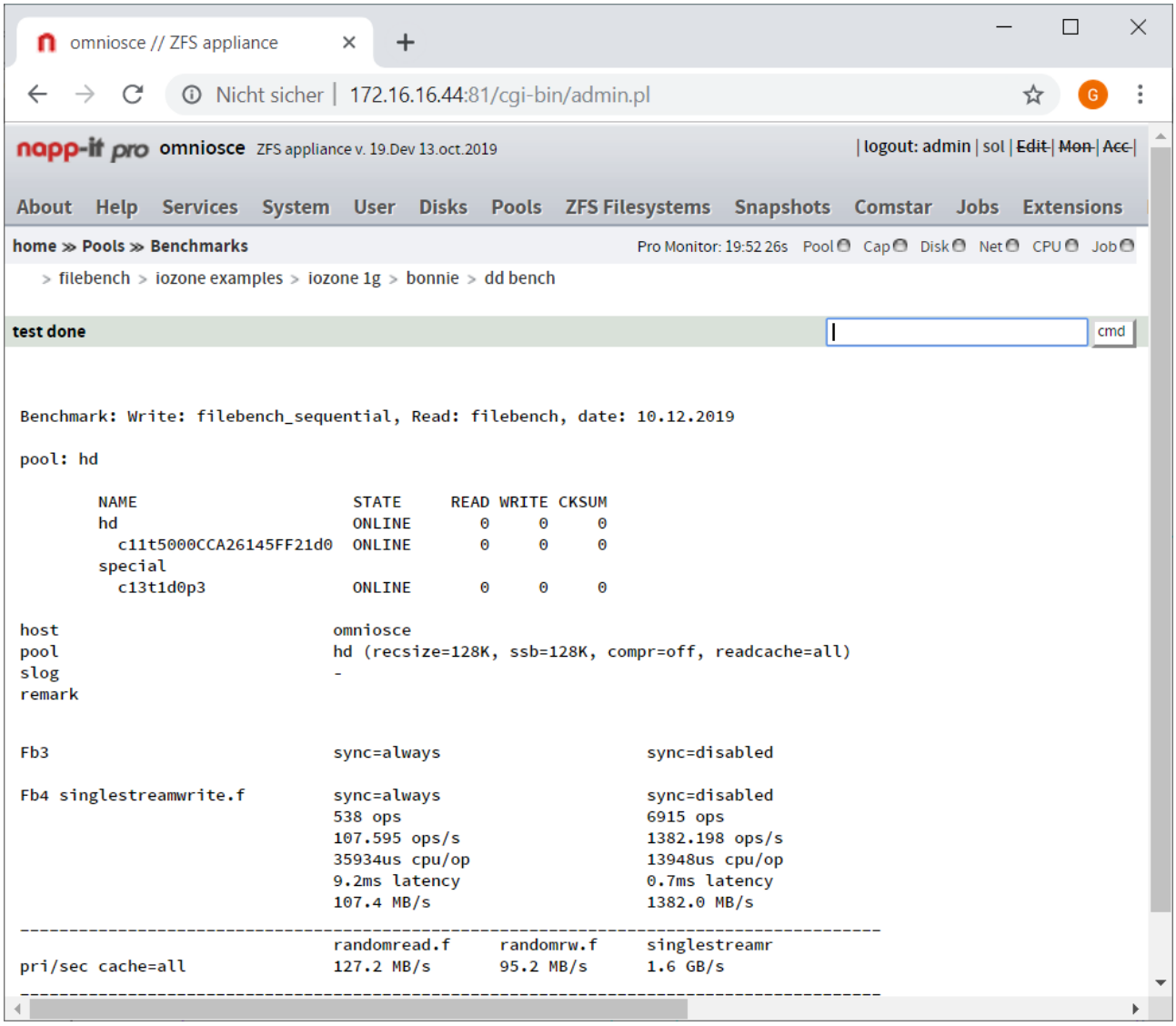
The above result is as expected for a single 12G SAS disk.
Randow rw is 163 MB/s, singlestreamwrite with the help of cache is 445 MB/s and sequential sync write is low with 35 MB/s.

3. Effect of adding a special vdev (Intel Optane 900p1 in pass-through mode) special_small_blocks=0



Result compared to 1.) is not so different.
Main advantage now is that metadata is on the Optane but as this is cached by Ram (Arc) there may be a difference only in a multiuser environment with a lot of random data.

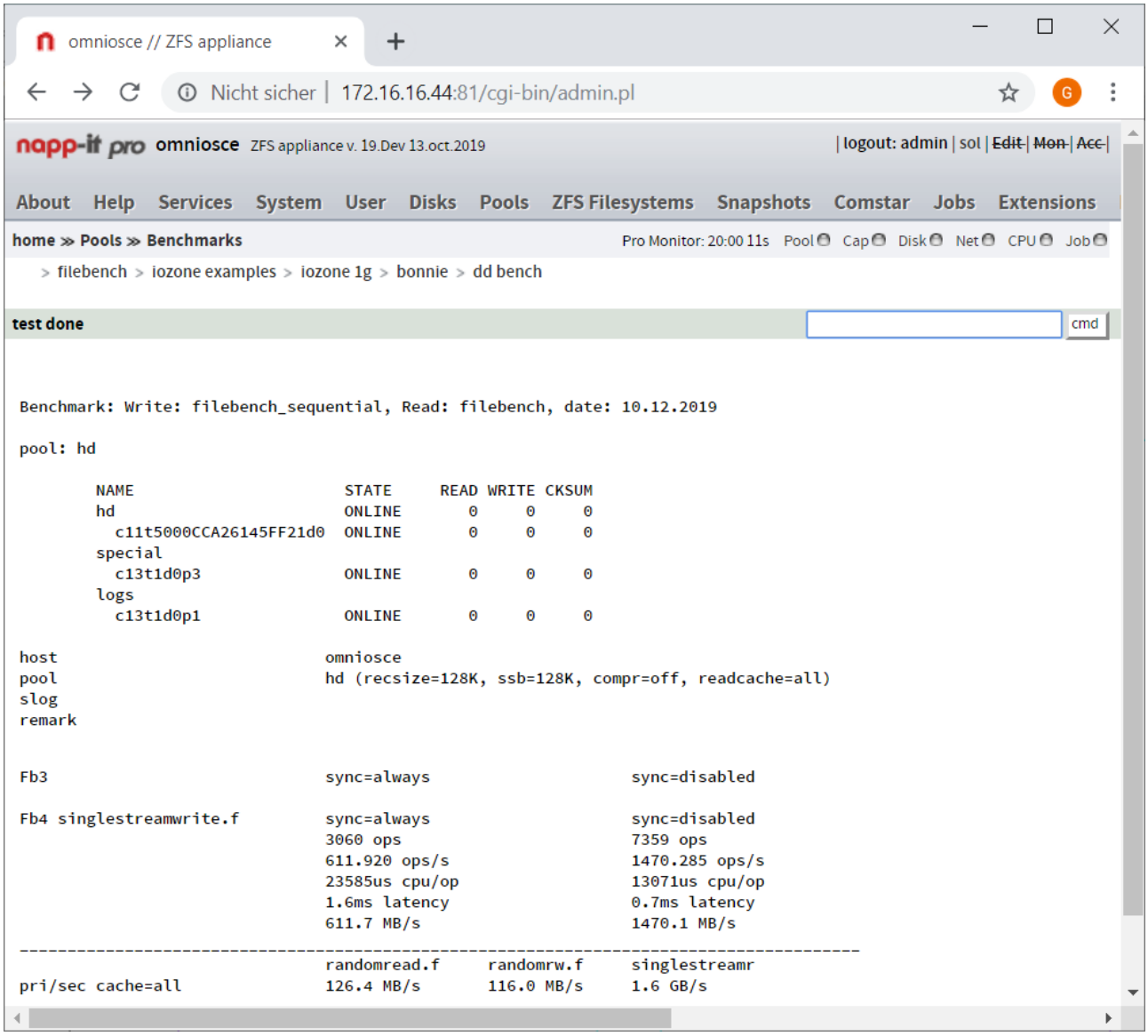
4. Effect of adding a special vdev (Intel Optane 900p1 in pass-through mode) special_small_blocks=128K and recsize also 128K



Result compared to 2.) is very different.
As the filesystem recordsize is equal to the special_small_block size, all data land on the Optane. This is why you want this feature, to decide if a filesystem writes to regular vdevs or the special vdev.

Sync write performance doubles (using onpool ZIL) and random write performance is around 3x better.

5. Effect of adding a special vdev (Intel Optane 900p1 in pass-through mode) + Slog special_small_blocks=128K and recsize also 128K



Result compared to 3.) shows clearly
Sync write performance with an additional Slog (Optane) goes up from around 100 MB/s to over 600 MB/s

First result:

- A special vdev that only holds metadata can help in some situations
- A special vdev to store files for single ZFS filesystems based on recordsize can be a huge improvement
- A special vdev does not replace an Slog.

- Sequential and random performance read/write (sync disabled) jumps from slow disk to fast NVMe performance.

6. What happens with a faster pool from multi-mirror and a P3600 as special vdev?

In the first round we have used a slow pool (single disk) and the performance improvement was dramatical for a filesystem using the Optane instead the disk. What happens with a faster pool?

Benchmark: Write: filebench_sequential, **Read:** filebench, **date:** 10.24.2019

pool: av

```
        NAME                      STATE     READ WRITE CKSUM
        av                        ONLINE       0     0     0
          mirror-0                ONLINE       0     0     0
            c0t5000CCA36ACBB845d0 ONLINE       0     0     0
            c0t5000CCA36ACFD81Fd0 ONLINE       0     0     0
            c0t5000CCA36AD1A823d0 ONLINE       0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000CCA36ACE362Ed0 ONLINE       0     0     0
            c0t5000CCA36ACE49E9d0 ONLINE       0     0     0
            c0t5000CCA36ACED5FEd0 ONLINE       0     0     0
          mirror-2                ONLINE       0     0     0
            c0t5000CCA36ACE89F3d0 ONLINE       0     0     0
            c0t5000CCA36ACF40F7d0 ONLINE       0     0     0
            c0t5000CCA36ACFD81Dd0 ONLINE       0     0     0
          mirror-3                ONLINE       0     0     0
            c0t5000CCA36ACE9172d0 ONLINE       0     0     0
            c0t5000CCA36ACED692d0 ONLINE       0     0     0
            c0t5000CCA36ACED794d0 ONLINE       0     0     0
          mirror-4                ONLINE       0     0     0
            c0t5000CCA36ACE9A58d0 ONLINE       0     0     0
            c0t5000CCA36ACE9A59d0 ONLINE       0     0     0
            c0t5000CCA36ACE9BC6d0 ONLINE       0     0     0
        spares
          c2t2d0                  AVAIL
```

host	**av-ablage**
pool	**av** (recsize=128K, ssb=-, compr=off, readcache=all)
slog	-
remark	

Fb3	**sync=always**	**sync=disabled**
Fb4 singlestreamwrite.f	**sync=always**	**sync=disabled**
	186 ops	5056 ops
	37.198 ops/s	1011.095 ops/s
	97408us cpu/op	37681us cpu/op
	26.7ms latency	1.0ms latency
	37.0 MB/s	**1010.9 MB/s**

	randomread.f	randomrw.f	singlestreamr
pri/sec cache=all	**125.4 MB/s**	**210.5 MB/s**	**1.1 GB/s**

Now we have around 1 GB/s read, randomread > 100 MB/s and randomrw > 200 MB/s
What happens to this pool if we add an Intel P3600 400GB as a special vdev to this pool?

7. Fast pool with special vdev

Pool with multi-mirror and Intel P3600- 400 as special vdev

```
Benchmark: Write: filebench_sequential, Read: filebench, date: 10.28.2019

pool: av

        NAME                        STATE     READ WRITE CKSUM
        av                          ONLINE       0     0     0
          mirror-0                  ONLINE       0     0     0
            c0t5000CCA36ACBB845d0   ONLINE       0     0     0
            c0t5000CCA36ACE362Ed0   ONLINE       0     0     0
            c0t5000CCA36ACE49E9d0   ONLINE       0     0     0
          mirror-1                  ONLINE       0     0     0
            c0t5000CCA36ACE89F3d0   ONLINE       0     0     0
            c0t5000CCA36ACE9172d0   ONLINE       0     0     0
            c0t5000CCA36ACE9A58d0   ONLINE       0     0     0
          mirror-2                  ONLINE       0     0     0
            c0t5000CCA36ACE9A59d0   ONLINE       0     0     0
            c0t5000CCA36ACE9BC6d0   ONLINE       0     0     0
            c2t2d0                  ONLINE       0     0     0
          mirror-3                  ONLINE       0     0     0
            c0t5000CCA36ACED5FEd0   ONLINE       0     0     0
            c0t5000CCA36ACED692d0   ONLINE       0     0     0
            c0t5000CCA36ACED794d0   ONLINE       0     0     0
          mirror-4                  ONLINE       0     0     0
            c0t5000CCA36ACF40F7d0   ONLINE       0     0     0
            c0t5000CCA36ACFD81Dd0   ONLINE       0     0     0
            c0t5000CCA36ACFD81Fd0   ONLINE       0     0     0
        special
          mirror-6                  ONLINE       0     0     0
            c20t1d0                 ONLINE       0     0     0
            c21t1d0                 ONLINE       0     0     0
        spares
          c0t5000CCA36AD1A823d0     AVAIL

host                            av-ablage
pool                            av (recsize=128K, ssb=128K, compr=off, readcache=all)
slog                            -
remark


Fb3                             sync=always                     sync=disabled

Fb4 singlestreamwrite.f         sync=always                     sync=disabled
                                509 ops                         4056 ops
                                101.787 ops/s                   811.147 ops/s
                                134391us cpu/op                 42088us cpu/op
                                9.8ms latency                   1.2ms latency
                                101.6 MB/s                      810.9 MB/s
-------------------------------------------------------------------------------------
                                randomread.f    randomrw.f      singlestreamr
pri/sec cache=all               89.8 MB/s       150.4 MB/s      946.9 MB/s
-------------------------------------------------------------------------------------
```

What's going on?

With the Intel P3600 as special vdev, performance go down from around 1000 MB/s to 800 MB/s. Randomread from 125 MB/s to 90 MB/s, randowrw from 210 MB/s to 150 MB/s and even singlestreamread (that is often ram-cache performance) go down from 1100 MB/s to 950 MB/s.

So let's do more tests

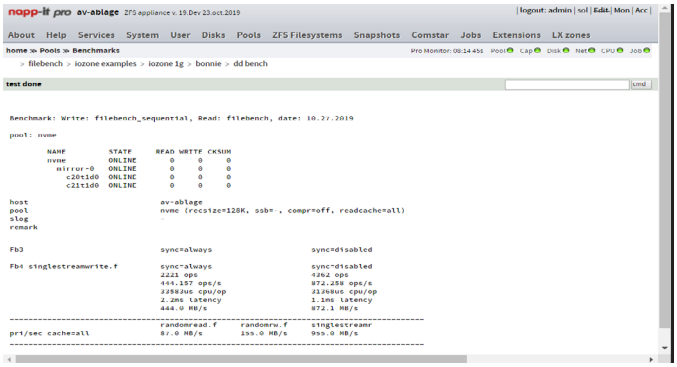
Benchmark with special vdev and small block size=0

```
Benchmark: Write: filebench_sequential, Read: filebench, date: 10.31.2019

pool: av

        NAME                       STATE     READ WRITE CKSUM
        av                         ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000CCA36ACBB845d0  ONLINE       0     0     0
            c0t5000CCA36ACE362Ed0  ONLINE       0     0     0
            c0t5000CCA36ACE49E9d0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000CCA36ACE89F3d0  ONLINE       0     0     0
            c0t5000CCA36ACE9172d0  ONLINE       0     0     0
            c0t5000CCA36ACE9A58d0  ONLINE       0     0     0
          mirror-2                 ONLINE       0     0     0
            c0t5000CCA36ACE9A59d0  ONLINE       0     0     0
            c0t5000CCA36ACE9BC6d0  ONLINE       0     0     0
            c2t2d0                 ONLINE       0     0     0
          mirror-3                 ONLINE       0     0     0
            c0t5000CCA36ACED5FEd0  ONLINE       0     0     0
            c0t5000CCA36ACED692d0  ONLINE       0     0     0
            c0t5000CCA36ACED794d0  ONLINE       0     0     0
          mirror-4                 ONLINE       0     0     0
            c0t5000CCA36ACF40F7d0  ONLINE       0     0     0
            c0t5000CCA36ACFD81Dd0  ONLINE       0     0     0
            c0t5000CCA36ACFD81Fd0  ONLINE       0     0     0
        special
          c20t1d0                  ONLINE       0     0     0
          c21t1d0                  ONLINE       0     0     0
        spares
          c0t5000CCA36AD1A823d0    AVAIL

host                          av-ablage
pool                          av (recsize=128K, ssb=0, compr=off, readcache=all)
slog                          -
remark


Fb3                           sync=always                    sync=disabled

Fb4 singlestreamwrite.f       sync=always                    sync=disabled
                              173 ops                        4773 ops
                              34.599 ops/s                   954.415 ops/s
                              162416us cpu/op                44592us cpu/op
                              28.6ms latency                 1.0ms latency
                              34.4 MB/s                      954.2 MB/s
-------------------------------------------------------------------------------------
                              randomread.f     randomrw.f    singlestreamr
pri/sec cache=all             86.0 MB/s        141.2 MB/s    891.0 MB/s
-------------------------------------------------------------------------------------
```

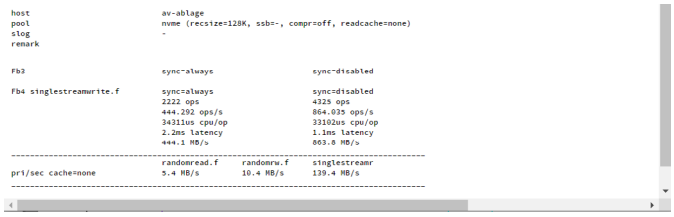
As espected.
Values not as good as when the filesystem is forced to use the special vdev and even not as good than without special vdev. Metadata only on a special vdev does not help on benchmarks with a quite empty pool. From expectation, this may change with a quite full pool and a lot of metadata not in cache.

8. Pool with NVMe vs special vdev

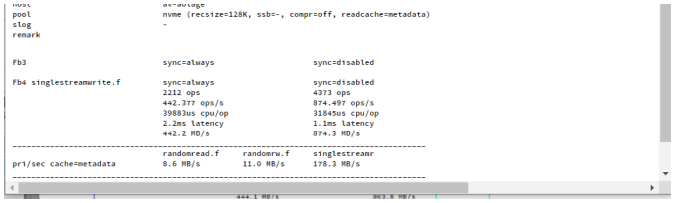
same NVMe than the former special vdev



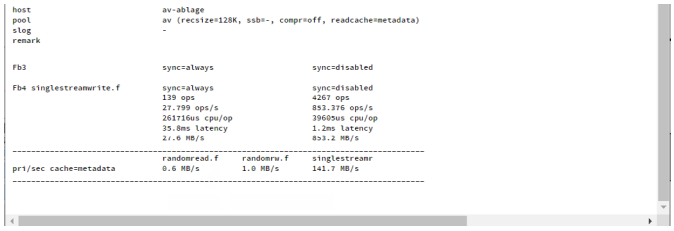
NVMe cache=none



NVMe cache=metadata



Raid-10 cache= metadata



9. Compare results

Raid 10 (see 6.) vs special vdev vs NVMe Pool

Result Raid-10 Pool	Raid-10 (cache=all)	Raid-10 (cache=meta)	Raid-10 special, cache=all	Raid-10 special, cache=all, Slog
9.1 singlestreamwrite sync	37 MB/s	27 MB/s	107 MB/s	611 MB/s
9.2 singlestreamwrite async	1010 MB/s	853 MB/s	1382 MB/s	1470 MB/s
9.3 randomread	125 MB/s	0,6 MB/s	127 MB/s	126 MB/s
9.4 randomread/write	210 MB/s	1,0 MB/s	95 MB/s	116 MB/s
9.5 singlestreamread	1,1 GB/s	141 MB/s	1,6 GB/s	1,6 GB/s

Result NVMe Pool	NVMe (cache=all)	NVMe (cache=meta)	NVMe (cache=none)
9.6 singlestreamwrite sync	444 MB/s	442 MB/s	444 MB/s
9.7 singlestreamwrite async	872 MB/s	874 MB/s	863 MB/s
9.8 randomread	87 MB/s	8,6 MB/s	5,4 MB/s
9.9 randomread/write	155 MB/s	11 MB/s	10,4 MB/s
9.10 singlestreamread	995 MB/s	178 MB/s	139 MB/s

9.11 Fast disk pool + Optane 900

What happens when you add an Optane to the game

Disk pool with P3600 as special vdev and Optane Slog

```
Benchmark: Write: filebench_sequential, Read: filebench, date: 10.28.2019

pool: av

        NAME                      STATE     READ WRITE CKSUM
        av                        ONLINE       0     0     0
          mirror-0                ONLINE       0     0     0
            c0t5000CCA36ACBB845d0 ONLINE       0     0     0
            c0t5000CCA36ACE362Ed0 ONLINE       0     0     0
            c0t5000CCA36ACE49E9d0 ONLINE       0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000CCA36ACE89F3d0 ONLINE       0     0     0
            c0t5000CCA36ACE9172d0 ONLINE       0     0     0
            c0t5000CCA36ACE9A58d0 ONLINE       0     0     0
          mirror-2                ONLINE       0     0     0
            c0t5000CCA36ACE9A59d0 ONLINE       0     0     0
            c0t5000CCA36ACE9BC6d0 ONLINE       0     0     0
            c2t2d0                ONLINE       0     0     0
          mirror-3                ONLINE       0     0     0
            c0t5000CCA36ACED5FEd0 ONLINE       0     0     0
            c0t5000CCA36ACED692d0 ONLINE       0     0     0
            c0t5000CCA36ACED794d0 ONLINE       0     0     0
          mirror-4                ONLINE       0     0     0
            c0t5000CCA36ACF40F7d0 ONLINE       0     0     0
            c0t5000CCA36ACFD81Dd0 ONLINE       0     0     0
            c0t5000CCA36ACFD81Fd0 ONLINE       0     0     0
        special
          mirror-6                ONLINE       0     0     0
            c20t1d0               ONLINE       0     0     0
            c21t1d0               ONLINE       0     0     0
        logs
          c22t1d0                 ONLINE       0     0     0
        spares
          c0t5000CCA36AD1A823d0   AVAIL

host                            av-ablage
pool                            av (recsize=128K, ssb=128K, compr=off, readcache=all)
slog
remark

Fb3                             sync=always                     sync=disabled

Fb4 singlestreamwrite.f         sync=always                     sync=disabled
                                3070 ops                        3828 ops
                                613.797 ops/s                   765.526 ops/s
                                65548us cpu/op                  74822us cpu/op
                                1.6ms latency                   1.3ms latency
                                613.6 MB/s                      765.3 MB/s
-------------------------------------------------------------------------------
                                randomread.f    randomrw.f      singlestreamr
pri/sec cache=all               91.4 MB/s       146.1 MB/s      965.2 MB/s
-------------------------------------------------------------------------------
```

Disk pool with Optane as special vdev and Slog

```
Benchmark: Write: filebench_sequential, Read: filebench, date: 10.28.2019

pool: av

        NAME                      STATE     READ WRITE CKSUM
        av                        ONLINE       0     0     0
          mirror-0                ONLINE       0     0     0
            c0t5000CCA36ACBB845d0 ONLINE       0     0     0
            c0t5000CCA36ACE362Ed0 ONLINE       0     0     0
            c0t5000CCA36ACE49E9d0 ONLINE       0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000CCA36ACE89F3d0 ONLINE       0     0     0
            c0t5000CCA36ACE9172d0 ONLINE       0     0     0
            c0t5000CCA36ACE9A58d0 ONLINE       0     0     0
          mirror-2                ONLINE       0     0     0
            c0t5000CCA36ACE9A59d0 ONLINE       0     0     0
            c0t5000CCA36ACE9BC6d0 ONLINE       0     0     0
            c2t2d0                ONLINE       0     0     0
          mirror-3                ONLINE       0     0     0
            c0t5000CCA36ACED5FEd0 ONLINE       0     0     0
            c0t5000CCA36ACED692d0 ONLINE       0     0     0
            c0t5000CCA36ACED794d0 ONLINE       0     0     0
          mirror-4                ONLINE       0     0     0
            c0t5000CCA36ACF40F7d0 ONLINE       0     0     0
            c0t5000CCA36ACFD81Dd0 ONLINE       0     0     0
            c0t5000CCA36ACFD81Fd0 ONLINE       0     0     0
        special
          c22t1d0p2               ONLINE       0     0     0
        logs
          c22t1d0p1               ONLINE       0     0     0
        spares
          c0t5000CCA36AD1A823d0   AVAIL

host                            av-ablage
pool                            av (recsize=128K, ssb=128K, compr=off, readcache=all)
slog
remark

Fb3                             sync=always                     sync=disabled

Fb4 singlestreamwrite.f         sync=always                     sync=disabled
                                2900 ops                        7445 ops
                                579.953 ops/s                   1488.894 ops/s
                                67060us cpu/op                  67222us cpu/op
                                1.7ms latency                   0.7ms latency
                                579.8 MB/s                      1488.7 MB/s
-------------------------------------------------------------------------------
                                randomread.f    randomrw.f      singlestreamr
pri/sec cache=all               86.6 MB/s       109.2 MB/s      949.0 MB/s
-------------------------------------------------------------------------------
```

To weight these values, compare a pure Optane basic pool

```
Benchmark: Write: filebench_sequential, Read: filebench, date: 10.28.2019

pool: optane

        NAME         STATE     READ WRITE CKSUM
        optane       ONLINE       0     0     0
          c22t1d0p2  ONLINE       0     0     0

host                            av-ablage
pool                            optane (recsize=128K, ssb=-, compr=off, readcache=all)
slog                            -
remark

Fb3                             sync=always                     sync=disabled

Fb4 singlestreamwrite.f         sync=always                     sync=disabled
                                3542 ops                        6364 ops
                                707.761 ops/s                   1272.676 ops/s
                                30119us cpu/op                  63216us cpu/op
                                1.4ms latency                   0.8ms latency
                                707.6 MB/s                      1272.5 MB/s
-------------------------------------------------------------------------------
                                randomread.f    randomrw.f      singlestreamr
pri/sec cache=all               88.6 MB/s       113.8 MB/s      942.8 MB/s
-------------------------------------------------------------------------------
```

The results:

Sync write
There is a huge bost on sync write if you add an Optane Slog
A large NVMe special vdev and a small Optane Slog gives perfect sync performance, similar to a pure Optane pool.

Async Write:
Equal to the special vdev

Random Access:
Quite similar with Optane and P3600 special vdev

10. Conclusion

Due the low number of tests, I will concentrate on important effects

10.1. Performance of a diskbased Raid-10 pool depends on its raw performance but can be improved massively by the rambased read/write caches of ZFS (see 9.1).

The Multi-Raid-10 diskpool is faster on writes than the NVMe pool (beside sync).
Read performance depends massively on RAM caching. Without a cache performance is worse.

This is a known fact. Ram for caching on ZFS can improve performance of slow pools massively but depend on cache hits on reads. Sync write performance is really bad without Slog.

10.2 If you add and use an NVMe as special vdev for a filesystem, its async write performance is even better than the performance of a pool from same NVMe. (9.2 vs 9.7)

Sync write is medium. This indicates that the Zil logging is spread over the whole pool (9.1 vs 9.6) but sync performance is still much better than the pool alone (2x to 4x).
An additional Slog (Optane) improves sync performance to 10x -20x.
Random read and write to the special vdev vs NVMe is not consistent. (9.3-5 vs 9.8-10)
but the filesystem on a special vdev paired with the pool performes phantastic.

So from a first view, you may asume that a special vdev is worse compared to the performance improvements due rambased read/ write caching. A more deeper view explains the difference.

10.3 What are the problems of a disk based pool?
Why a special vdev is good despite

- Concurrent read/write on disks but also traditional Flash (beside Optane) can reduce pool performance massively. As for every read/write you must read metadata first, outsourcing metadata from the pool can give a performance improvement especially on high load systems as it reduces regular pool access.

- Only a small part of metadata is in cache.
The Arc/L2Arc caches metadata and small random reads on a most accessed/last accessed basis. It does not cache whole files or sequential data. First access to metadata is always slow and access to your file-system is only improved by caches for most active data.

If you asume that 1% of your data is metadata, a pool with 100 TB size would require a cache of 1 TB to hold all metadata so this is not a solution even if you asume a persistent cache. A special vdev for meta-data (1% poolsize) is the solution for this problem.

- If you enable dedup, you can asume that a dedup table can be up to 5% of dedup data. Up to now, this should be RAM and this RAM reduces amount of Arc cache or write cache. Outsourcing the dedup table to a high performance NVMe (Optane) gives you dedup
performance and all RAM remain available for read/write caching and its performance improvement.

- Performance of a ZFS pool is inconsistent and not predictable
First access is always slow. RAM does not help. This is where special vdevs where you force some filesys-tems to the special vdev can help. While there is no further cache improvement, all read/write accesses happen with the raw performance of the special vdev.

11. When I would expect a special/dedup to be helpful or very helpful

11.1. Large pool with volatile random data access patterns
Access to any metadata on a special vdev is fast what makes the pool more responsive.

11.2 Shorter resilver time
A resilver needs to read all metadata. Faster access to metadata on a special vdev means shorter resilver time.

11.3 Guaranteed performance even on first access for single filesystems on special vdevs.
For databases or VM storage you may want a guaranteed cache independent performance. Filesystems on special vdevs can guarantee this.

11.4 Sync Write performance on selected filesystems without Slog is much better (2x-4x in my tests) than on the disk pool but far below the results with a dedicated Slog (Optane, 10-20x).

12. When a special vdev is not needed or helpful

12.1. Sequential data access (ex mediaserver)
Performance is pool limited. Neither RAM nor a special vdev really helps.

Nearly always a disk pool is sequentially faster on async writes and reads than your network

12.2 Mixed access patterns but a constant amount of active data.
Rambased caching can improve performance more than a special vdev.
ex: SoHo filer with few users only or an Office filer with limited number of active files.

13. General suggestion

A special vdev NVMe allows a performance jump for large disk pools in general (metadata access) or can give full NVMe performance for selected filesystems (read and async write, improved sync write).

If you really need fast sync write, use an additional Slog (Optane, WD SS530 etc)

A special vdev should have powerloss protection (or at least a decent powerloss behaviour like the non datacenter Optane) as it holds critical data.

A special vdev should provide a similar redundancy level as the pool (2/3-way mirror). With several mirrors, capacity for special vdevs increases and load is spread over them.

Fazit

A huge and cheap disk pool paired with affordable SSD/12G SAS/NVMe as special vdev mirrors for metadata and selected filesystems + a small Slog (ex 4801x-100, WD SS530) allows to build a single multi purpose pool that offers capacity and superiour performance when needed at a decent price.

more manuals:
https://napp-it.org/manuals/index_en.html
Slog performance (Intel Optane)
https://napp-it.org/doc/downloads/optane_slog_pool_performane.pdf