napp-it ZFS Storage principles
for high performance SMB fileservices

Server: Dual Xeon, 8/40 GB RAM
Intel X540, mtu 9000
(JumboFrames)

Server OS: Oracle Solaris 11.3
SMB 2.1
OmniOS 151017
SMB 2.1

Client OS: MacPro OSX 10.10/10.11
with Sanlink2

Client OS: Win 8.1/ 10 Pro
with Intel X540-T

Questions: SMB1 vs SMB2 vs NFS
mtu 1500 vs MTU 9000
NVMe vs SSD vs Disks
Raid-0 vs Mirror vs RaidZ

Single user vs multiple user
Single pool vs multi pools

Solaris 11.3 vs new OmniOS
8 GB RAM vs 40 GB RAM

Dedupe and LZ4

NVMe vs Enterprise SSD
vs regular SSD

Write behaviour of them
vs filesize over time

 OSTuning options

RAM requirement
Client

Hardware: Apple MacPro, OSX 10.10
4 Core 3.7 GhZ, 12 GB RAM
Network: 10 Gbase-T Sanlink 2

Local disk performance on Apple NVMe that is quite similar to an Samsung 950 Pro M2

Performance:
write 994 MB/s
read 961 MB/s

Switch
HP 5900, 48 x 10 Gbase-T

Storage
Dual Xeon, 40 GB RAM. Intel X540 10Gbase-T and Jumboframes enabled on all tests

Server OS (Round 1)
Oracle Solaris 11.3 as it offers SMB 2.1.
SMB1, 10 GbE (Solaris)
Pool Layout: 3 x Mirror of 2 TB HGST 7k U/m

SMB1, MTU 1500, 10 GbE (connect to cifs://)

SMB 1, MTU 9000, 10 GbE (connect to cifs://)
SMB 2.1, 10 GbE (Solaris)
Pool Layout: 3 x Mirror of 2 TB HGST 7k U/m

SMB 2.1, MTU 1500, 10 GbE (connect to smb://)
SMB 2.1 gives a boost over SMB1 and a second boost when using JumboFrames

SMB 2.1, 10 GbE
Pool Layout: 7 x Raid-0 of 2 TB HGST 7k U/m

SMB 2.1, MTU 9000, 10 GbE (connect to smb://)
SMB 2.1, MTU 9000, 10 x HGST 2TB
Pool Layout: 10 x Raid-Z2 of 2 TB HGST 7k U/m

SMB 2.1, MTU 9000, 7 x HGST 2TB raid-0
SMB 2.1, MTU 9000, 10 x HGST 2TB
Raid Z2, single vdev

There is no advantage of the raid-0 with a much higher iops rate (7 x a single disk vs 1 x a single disk)
Pool Layout: 10 x Raid-Z2 of 2 TB HGST 7k U/m (Solaris)
Windows 8.1 pro, 10 GbE, MTU 1500

On Windows, MTU 1500 is weak on reads

Hardware is a 6 core Xeon with 32 GB RAM, an Intel X540-T1 and a local Sandisk Extreme Pro SSD

OSX 10.9, MTU 1500

There seems to be a problem with OSX 10.9 and MTU 9000 as reads are low even with MTU 9000

I made an update from 10.9 to 10.11 but results were the same.
A clean install of OSX 10.11 fixes this problem.
SMB 2.1, 10 GbE, MTU 9000 (Solaris)
Pool Layout: Single NVMe Intel P750

SMB 2.1, MTU 9000,
1 x NVMe Intel p750-400

Performance degradation is noticeable with a Raid-Z2 pool from 10 spindels.
Write values for 6 concurrent users jumps between 130 MB and 200 MB/s with reads always 300 MB/s and better
SMB 2.1, 10 GbE, MTU 9000 (Solaris)
Single User on 1 disk NVMe Intel P750

The performance degradation with 6 users on writes is lower due to the higher iops of the NVMe. Read degradation is quite similar to the spindle based Z2 pool. The ZFS Ram cache seems to equal this.

SMB 2.1, 10 GbE, MTU 9000, Multipool-Test vs Z2
Single User on 1 disk NVMe Intel P750

This test is running with 5 user on Raid-Z2, concurrently to the left single user.

Two pools accessed at the same time
The left NVMe pool runs with quite full speed concurrently to the Z2 pool with 5 users.
Round 2: SMB2 on OmniOS bloody 151017 beta (Dez 2015)

During this tests, the new OmniOS with SMB2.1 support arrived in a beta state. I installed the OmniOS 151017 beta to the same hardware with same settings. After the first test it seems that it behaves different to Solaris. While I saw the same behaviour like a good performance on some config and a very bad performance on other. But some configurations that were fine with Solaris are bad with OmniOS and vice versa:

MacPro, OSX 10.5, Raid-Z2, MTU 9000, SMB 2.1  
Values on Solaris were 500 MB/s and 800 MB/s  
Values on Solaris were 520 MB/s and 101 MB/s

I updated all MacPros to 10.11.2 and Windows on MacPros with newest Sanlink driver for further tests  
after this, I saw a consistent high write and read performance with OSX and good values on Windows

Pool Layout: 10 x Raid-Z2 of 2 TB HGST 7k U/m  
good values, slightly slower than Solaris 11.3 with 501 MB/s write and 836 MB/s read

Pool Layout: 1 x basic Intel P750 NVMe  
similar on NVMe compared to Solaris with 605 MB/s write and 830 MB/s read
MacPro, OSX 10.5, Raid-Z2, MTU 9000, SMB 2.1 (OmniOS) Concurrent access: 1 user to another NVMe pool
5 concurrent user/ MacPros

Same like on Solaris
With many user working on the same pool, especially writes values are quite low
Another user that is working on a different pool das better values so pool is more a limiting factor than network

MacPro with Win 8.1, Raid-Z2, MTU 9000, 1 user
older Promise Sanlink 2 driver (Begin 2015)

Same MacPro with Windows 8.1
but with newest Sanlink2 driver Nov. 2015

A newer driver can double read values. This is similar to OSX where another OS release can have the same effect.
In general Windows is lightly slower than OSX with default settings. Solaris and OmniOS perform similar.

Optimal settings X540
Interrupt throttling: deactivated
Jumbo Packet: 9014

use newest Intel driver:
(Sanlink2: Nov 2015)
Round 3 - What happens when I reduce Server RAM from 40 GB to 8 GB on same hardware and sequential tests

**Single user access to the Raid-Z2**
10 x HGST 2 TB with 8GB RAM

**Single user access to the raid-0**
3 x Intel s 3610-1,6 TB

With a single user/ single thread sequential performance with only 8 GB server-RAM is about 30% lower than with 40 GB RAM but more than enough for most workloads and 4x to 6x to what's possible with 1G networks.

**Six concurrent MacPro/user access to the Raid-Z2**
10 x HGST 2 TB with 8GB RAM

**Six concurrent MacPro/user access to the raid-0**
3 x Intel SSD S3610-1,6 TB

With only 8 GB RAM, many reads must come from disk. With the limited iops of the spindle Z2 pool performance degradation especially on writes is quite massive. The SSD pool with a much higher iops rate is more than twice as fast but only half as fast as on same tests with 40 GB RAM.
Round 5: Some tests

Dedupe and Compress (Tests run against an Intel P750-400)
Referenz: dedup and Compress off (469/708 MB/s)

Compress LZ4 enabled (473/687 MB/s)

Dedupe enabled (380 / 652 MB/s)

Dedupe and LZ4 enabled (355 / 647 MB/s)

There is a slight performance degradation with compress and dedupe. But while dedupe has a massive impact on RAM needs (count up to 5 GB RAM per TB data + RAM needs for ARC cache), LZ4 is a good idea even on data without a good compressability.
Round 6: OS tuning options

Solaris and OmniOS are very fast but defaults are more optimized for 1G networks.

Let's try some tunings (Napp-it 16.01pro, menu System > System Tuning) to a single Intel P750-400

NFS results on OSX with "napp-it default tuning" enabled

For whatever reason, default NFS performance on OSX is weak

Result:

With default OmniOS settings, NFS performance is bad but with the default tuning options in napp-it: much better
SMB2 results on OSX

default OmniOS settings
Result: similar read values but much better write values

OmniOS with „napp-it basic tuning“ enable

SMB2 results on Windows 10 (MacPro via Bootcamp)

default OmniOS settings,
Windows MTU 1500 and int_throtteling disabled

default OmniOS settings
MTU 9000 and int_throtteling disabled

second check with OS defaults

I made a second check with OS-Default.
With MTU=1500 and MTU=900 I got bad readvalues.
I was not able to get the same values than above.
This seems to be related to a weak cable !!!!
Another cable fixes the problem
The values with 867 MB write/s and 660 MB/s with
- OmniOS base tunings
- Jumboframes enabled
- int_throtteling disabled on Windows are always perfect, does not matter if I reboot or switch settings

OmniOS basic tuning
Windows MTU 1500, and int_throtteling disabled

OmniOS basic tuning
Windows MTU 9000, and int_throtteling disabled
Single NVMe vs Enterprise 6G SSD vs Consumer 6G SSD vs 7k rpm 6G disk with large filesize

With the AJA testtool, you can select different video resolutions, compressors and file sizes where Aja writes a file per frame with a size of about 75 MB per frame with this resolution. The total file size determines the number of frames. I selected large 4k frames with uncompressed RGB16 and 64 GB file size to check behavior over time.

Intel P750 NVMe with large 4k framesize (75MB per frame)

Result:
Write transfer starts with full network performance of about 1000 MB/s and remains at that level with some minor drops.
Read performance remains constantly over 600 MB/s

![Image of Intel P750 NVMe with large 4k framesize]

Intel P750 NVMe with small PAL SD framesize (2,5 MB per frame)

Result:
During the write of nearly 7000 frames, write performance was constantly at an average of about 600 MB/s.
Read is constantly over 500 MB/s what shows the quality of the Intel NVMe disks regarding transfer rate and iops.

![Image of Intel P750 NVMe with small PAL SD framesize]
Intel S3610 SSD with large 4k framesize (75MB per frame)

Result:

Write transfer starts with full network performance of about 1000 MB/s and drops after some frames to about 500 MB/s.

The initial peak seems the effect of the ZFS write buffer.

Read performance remains constantly at 470MB/s after an initial lower value.

---

Intel S3610 SSD with small PAL SD framesize (2,5 MB per frame)

Result

During the write of nearly 7000 frames, write performance was constantly at an average of about 500 MB/s (near limit of 6G Sata)

Read remains constantly at about 450 MB/s
**Sandforce based SSD with large 4k framesize (75MB per frame)**

Result:

Write transfer starts with full network performance of about 1000 MB/s and drops after some frames to about 250 MB/s. The initial peak seems the effect of the ZFS write buffer.

Read performance remains constantly at about 470MB/s

---

**Sandforce based SSD with PAL SD framesize (2,5 MB per frame)**

Result:

During the write of nearly 7000 frames, write performance starts with over 500 MB/s and drops after some frames to about 360 MB/s

Read is constantly at about 450 MB/s .
Fazif of disk types:

Only NVMe and Enterprise SSD can offer a constant high write performance behavior. Older SSDs like the Sandforce based Winkom Pro are not as good but offer twice the write performance of disks.

While sequential performance with a single large file on a Raid array is quite similar between SSD and disk pools, the difference is huge when you check with many small files over time. Iops is the difference as an NVMe offers 100000 iops, Enterprise SSDs up to 40000, average SSDs up to 10000 and spindle disks around 100 iops under load.
Fazit

**SMB1 vs SMB 2.1**
SMB1 on OSX is very slow. SMB 2.1 is a must. With more than 1 GbE use SMB2+ on Windows as well.

**MTU 1500 vs MTU 9000 (JumboFrames)**
You should use Jumboframes for performance. Performance improvement is up to 20-50%

**SMB2 vs NFS**
I have had stability problems with NFS on OSX as disconnects happens as well as a very bad performance of NFS. Even with Jumboframes the best what I got without any tweaks was 70 MB/s write and 150 MB/s read. Together with the missing authentication or authorisation of NFS, this is not an option.

**RAID-0 vs Mirror vs Raid-Z**
For a single user test, differences are minimal as long as your pool is capable of a sequential performance > 1GB/s. In a multi-user/ multi-thread environment, there is a difference as iops performance of raid-Z is lower.

**NVMe vs SSD vs Spindle based pools**
For a single user test, differences are minimal as long as your pool is capable of a sequential performance > 1GB/s. In a multi-user/ multi-thread environment, there is a difference as iops of SSDs is much higher than with spindels. A single NVMe has nearly no advantage over 3 x 6G SSDs.

**Solaris 11.3 vs OmniOS 151017**
Solaris is on OSX slightly faster than the OmniOS beta. Results with Windows as client are similar.

**Windows 8.1/10 vs OSX 10.11**
I have not expected the result as OSX is slightly faster on 10G out of the box than Windows (same MacPro hardware). Other Windows results on a 6core Xeon PC (X540-T1) workstation are similar to the results on the MacPro with Windows. Deactivate Interrupt throttling on Windows and enable Jumboframes (X540 nic) is required.

**Server-RAM 40 GB vs 8 GB**
RAM is an important factor as reads from RAM-cache are 1000x faster than reads from disk. The reduced load on the disks also improves writes. The more concurrent user or processes the more important is RAM. For a single user 8GB is perfect even with 10GbE. For multiuser or multi process access, use much more RAM in a high performance setup.

**OS Tuning**
With a base tuning, see napp-it menu System > System tuning: can improve performance up to 20-30%

**In General**
In the last 10 years disk capacity grows from multi Gigabyte to multi Terabyte with a factor 1000 while network stay at 1Gb/s for years. If you need to copy or backup a disk or pool this can last days or over a week. Modern SSDs can give 10x the performance of 1 GB/s so you really need a 10 GB/s network.

10G Performance depend on settings like Jumboframes and SMB2+. Driver quality and general OS settings are essential. On some configs 10G performance is only minimal better than on an 1G network.

While 1G networks are at about 100 MB/s without special settings, the 10G is in the best case 6-8x as fast but only with settings like SMB2, Jumboframes or disabling interrupt throttling ex in X540 settings on Windows. I am under the impression that the OS defaults are more optimized for 1G than 10G performance.

**Conclusion:**
Use 10 GbE, SMB2+ and Jumboframes + some OS tunings if you need 10G performance. Care about OS version and drivers and about some driver settings like the irq throttling in X540 on Windows.

Oracle Solaris is out of the box slightly faster than current OmniOS beta. Both are a huge step forward compared to their predecessor especially on OSX or with 10 GbE.

For single large file and single user the sequential performance is ok with a pool from any sort of disks. If you need a really high continuous write performance with many small files, you must use enterprise class SSDs.

**Cabling with 10G Base-T is very critical.** Different cables can give 20-30% difference sometimes down to 1 Gb/s - ex 600MB/s write and 105 MB/s read, a new or shorter cable can fix the problem.
about using AJA or other benchmark tools

What to expect
Prior any test, calculate a thumb rule about expected values.
For example, a single harddisk cannot deliver more than 100-200 iops and 200-300 MByte/s
With small files in the outer tracks and a higher fragmentation expect 50-80 MByte/s
If values are outside of the expected range, you have a problem or you test cache qualities not disk qualities.

Cache/RAM effects
If you want to test overall storage quality, use all the RAM as readcache. This is where ZFS performance comes from.
as ZFS needs to calculate and process more data than other filesystems due the additional checksums. This is the
price of the extra security. Optionally enable sequential caching for L2ARC devices, If you want to compare your disks
only, not overall storage performance use only 2-4 GB RAM to reduce cache effects.
Usually disable sync-write and enable write back cache for iSCSI devices for best performance.
Enable sync or disable write back cache if you want to test performance on secure powerloss safe writes or a Slog.

Multiuser
Mostly you do not have only a single user on your storage that is viewing a video. For such a use case on
1G networks, any cheap NAS can give up to 100 MByte/s and you do not need high performance storage with
striped Raid and a lot of RAM as cache and CPU power. In such a case, data security is the only ZFS advantage.

With AJA, you can test effects of Multi-User use cases quite easily.
Create for example four shares on your pool, start AJA four times and connect each to a different share.
Then start a test on all four instances of AJA simultaniously, best with a large testfile size (16 or 64GB)
and compare results with single user values. Mostly you will find, that disk performance fall far below
your network performance. To reduce network effects you can use four client computers with a dedicated server nic.

File size
Some tests ex dd create a filestream to test a device without any parallel actions (queuedepth=1)
This is not a realworld scenario and can give the result that a harddisk is faster than a modern SSD.
Use also fileizes that are typical for your workload. Example if you are using the storage for maildate, you have many
small files. If you use the storage to edit videodata, you are more interested in the behaviour with large files.
In AJA you can simulate this with the videosize. If you enter a videosize of 720x576 PAL you create many small files
while a setting of 4k creates less but larger files. Use 4k videosize if you want to test sequential performance, use
PAL if you are interested in io behaviour with small files.

CPU effects
With AJA storage tests you should set 16 bit RGB as video codec
If you use a videocompressor, CPU performance of your client PC/desktop is part of the result.
If you enable compress or encryption ex on Solaris, CPU performance of the server is relevant

Performance over time
Especially SSDs are offering best iops performance. On desktop SSDs you find values up to 100000 on the
advertisements while enterprise SSDs offer more realistic values of 20000 - 80000 that they can keep under
constant load while cheap desktop SSDs can break down to 5000 iops after some time of usage.
If you want to test this effect, you need a test that runs for 1-2 hours and produces a
constant write load. You must check AJA graphs of performance over time

Fillrate and usage
Performance is quite often a function of fillrate and on SSDs of already used flash cells or overprovisioning.
If you want to compare results use empty disks and do a secure erase on SSDs prior tests

System settings of OmniOS/ Solaris
System settings are always an optimisation for a workload or a system/RAM setup. For an iSCSI, SMB or NFS
storage system test with RAM > 1-2GB, use the napp-it default tuning option in menu System to increase tcp buffers
as the defaults are optimized for very low RAM and use a current OmniOS with SMB 2.1 enabled (151017up).
LZ4 compress enabled gives mostly better performance values than compress disabled as read/writes are reduced.

Conclusion
If you want to compare two storage systems regarding these questions:
- use more that one AJA instance or use more clients simultaneously
- select a videosize according your usecase, ex 720 x 576 PAL for behaviour with small files
- select a large testfile ex 64GB for a long running test, check AJA read/write graph over time
- select a videocompressor to add some CPU load

Now you can compare the results of two storage systems or two storage layouts/ configurations.
Setup configuration ideas

For multi-user video-editing or other high performance lab setups you need high performance and/or high capacity. If you need both with the same storage, this can be quite expensive as you not only need the storage but also high performance disks and networking and can be very load (requires a server room). Some typical configurations:

1. High Performance (4k) lab or desktop storage, capacity up to several Terabyte for current working data.

You can use a silent desktop case with a 10 x Sata mainboard with up to seven PCI-e slots, example a SuperMicro X10SRL-F where you can insert for example 3 - 5 Intel P750 NVMe disks with up to 1,2 TB each. With a Raid-Z1 this gives you up to 4,8 TB of ultrafast storage, capable to serve several 4k video editing clients simultaneously with the same data.

As an additional option add up to 9 SSDs like 3,8TB Samsung PM 863 to onboard Sata what can give you up to 26 TB in a Raid-Z2 setup. Sata nr 10 is needed for your bootdisk.

Such a storage can be built ultra silent so you can place it beside the clients. If you use the remaining two slots for 2 x Dual 10G nics, you can connect 4 x 10G clients directly plus 2 x 1G clients or use one as uplink without the need of an extra expensive and loud 10G switch as you can enable bridgung between the nics. This will allow any client to see the server, each other and everything that is connected on the 1G uplink. Up from napp-it Pro 2016.05 dev you can enable the bridging function (OmniOS acts like a normal network switch) in menu System > Network Eth.

For more clients, you can use a 10G switch. As a future option there are the upcoming Intel X710 QSFP+ network adapters. They offer up to 2 x 40G QSFP+ or 8 x 10G SFP+ per nic. Drivers for OmniOS are on the way. With such a card, you can connect up to 8 x 10G clients per nic without the need of a switch. At the moment, QSFP+ optic tranceivers are quite expensive but cheap copper QSFP+ to 4 x SFP+ cables with up to 5m should be available soon.

2. Affordable very high capacity storage

You can use for example a 24 x 3,5” case like SuperMicro SC846BA-R1K28B with a a SuperMicro X10SRL-F and 3 x LSI/AVAGO/Broadcom HBAs with IT mode firmware like the model 9207-8i (expander/less, best for Sata disks). If you insert 24 x 10 TB disks in a 4 x 6 Raid Z2 or 20 disks in a 2 x 10 disk raid z2 (half the ipops) you achieve 160 TB usable. Add enough RAM and at least one NVMe as L2ARV and you have affordable but quite fast high capacity storage. This case cannot be placed near a desktop (too loud).

If capacity should go beyond, you can use cases like the SuperMicro https://www.supermicro.nl/products/chassis/4U/?chs=946 with expander and up to 90 SAS disks per case.

3. Affordable high performance/ high capacity storage

You can use a SuperMicro barebone like the http://www.supermicro.com/products/system/2U/2028/SSG-2028R-ACR24L.cfm with up to 24 x 2,5” disks. If you insert 20 x 3,8 TB SSDs like the Samsung PM 863 in a 2 x 10 SSD Raid-Z2 you have up to 60 TB ultrafast SSD only storage. This case cannot be placed near a desktop (too loud).

OSX 10.12 Sierra

I have repeated the benchmarks with newer OSX 10.11.5 or newest OSX 10.12 releases and found that 10G performance was not as good as before. I have tried some tuning recomendations like signing_required=no that are suggested on the internet but was not able to get an SMB performance better than around 200-300 MB/s - far below the former 600-800 MB/s.
**Effect of Readcache (Cache for Metadata and/or Data)**

ZFS is known to be fast with its advanced readcaches. How fast or slow is ZFS without caches?

For these tests, I use `filebench/singlstreamread` (napp-it menu Pools > Benchmark > Filebench). Caches can be enabled/disabled in napp-it menu Pools >

Filer with a 60% fill rate, 23% fragmentation, 2 x 6 disk RAID-z2 vdev of HGST HUS724040AL, filer under decent load from students during test. The term mb/s in filebench means MegaByte/s.

all caches on

| IO Summary: 157739 ops, 5257.764 ops/s, (5258/0 r/w), 5256.7mb/s, 211us cpu/op, 0.2ms latency |

19437: 34.518: Shutting down processes

all caches off

| IO Summary: 13757 ops, 458.554 ops/s, (459/0 r/w), 458.5mb/s, 2168us cpu/op, 2.2ms latency |

27837: 35.348: Shutting down processes read drops to 458 MB/s

**Now values of single disks**

A singlstreamread on a single disk HGSH HE8 Ultrastar

| IO Summary: 2554 ops, 85.130 ops/s, (85/0 r/w), 85.1mb/s, 3578us cpu/op, 11.7ms latency |

13519: 41.172: Shutting down processes

the same but with metadata caching on (data caching off)

| IO Summary: 2807 ops, 93.565 ops/s, (94/0 r/w), 93.5mb/s, 3317us cpu/op, 10.7ms latency |

4776: 41.229: Shutting down processes helps a little. With higher fragmentation, the difference may become bigger

and a single Intel S3610-480 (all caches off)

| IO Summary: 8874 ops, 295.792 ops/s, (296/0 r/w), 295.7mb/s, 2216us cpu/op, 3.4ms latency |

16725: 37.100: Shutting down processes

and finally a very old and slow WD green 1TB (all caches off)

| IO Summary: 94 ops, 3.133 ops/s, (3/0 r/w), 3.1mb/s, 67645us cpu/op, 304.0ms latency |

24202: 47.712: Shutting down processes ok.

now a low rpm WD RE4 (low power 2TB enterprise disks)

| IO Summary: 463 ops, 15.433 ops/s, (15/0 r/w), 15.4mb/s, 205954us cpu/op, 64.5ms latency |

24202: 47.712: Shutting down processes ok.
Result with low RAM

The result clearly shows that ZFS performance especially on read highly depend on its caches, probably more than on other filesystems. So especially with very low RAM and disks with very low iops especially read performance of ZFS seems really worse. The faster the disks the better the values. With readcache performance can increase dramatically. High iops is a key value for performance. With the WD green, results with 3,1 MB/s are dramatically bad while results with the newer HGST 8TB disk and 85MB/s and S3610 and 295 MB/s are ok. I am as well astonished about this very bad values on a slow disk with low iops values and cache disabled as mostly you do tests with all caches on to check overall system performance.

On writes, you always use the write cache that optimizes writes so default write values are much better. With sync=on you see similar weak write values as well with low iops disks. On a regular load, ZFS will hide these bad behaviour mostly with the readcache. So the result: avoid low iops disks and use as much RAM as possible for readcache.

Performance considerations regarding RAM: Summary

Aspect 1: sequential raw disk performance

This scales:
- on mirrors sequential read performance = n x number of disks
- on mirrors sequential write performance = n x number of vdevs
- on Raid-Z sequential read/write performance = n x number of datadisks
  ex a raid-z2 of 6 disks has 4 datadisks (150MB/s each) = 4 x 150 MB/s = 600 MB/s

Aspect 2: iops raw disk performance

This scales:
- on mirrors iops read performance = n x number of disks
- on mirrors iops write performance = n x number of vdevs
- on Raid-Z iops read/write performance = n x number of datadisks
  ex a dual raid-z2 pool (100 iops/ each) = 2 x 100 iops = 200 iops/s

Aspect 3: RAM effects

A Copy On Write (CoW) filesystem like ZFS does not place data sequentially on disk but spread them over the disk with a higher fragmentation than on non-CoW filesystems. This is the price of the much higher data security of ZFS. To compensate or overcompensate this, ZFS comes with one of the most advanced RAM based or SSD based cache strategies. With enough RAM most reads are delivered not from disk but from cache to avoid limited disk performances. On writes all small and slow random writes are collected in a rambased write cache and written after a few seconds as a large and fast sequential write.

RAM requirements:
- A Solarish OS requires around 2 GB RAM for stable operation with any poolsize
- 10% of RAM up to 4 GB RAM is used for the write cache
- RAM above is dynamically used as readcache (when not required otherwise) for metadata and small random reads.

total RAM suggestions:
A SoHo filer with a few users or a media server: 4-8 GB RAM
A Filer for VM use or more users 8-32 GB RAM

Special use cases:
If you want to cache nearly all data in RAM 1% of active data required to cache metadata + all random data. This can mean 32-256 GB RAM
A rule of thumb: 1-2 GB RAM/TB data

If you want to use realtime dedup add 2-5 GB RAM per TB dedup data
If you want to cache sequential video data add a fast NVMe as L2Arc and enable sequential caching
size of L2Arc should not exceed 5-10 x RAM

Attention: If you use an Slog device:
An Slog is not a write cache, but a logdevice for a secure write behaviour like to a cache+battery on hardwareraid