

OpenZFS: co je nového

Pavel Šnajdr
OpenAlt 2017



OpenZFS: co je nového

- stable release 0.7.0
- Encryption
- dRAID
- Sequential Scrubs and Resilverers
- VDEV Removal
- ZFS Channel Programs
- Allocation Classes

OpenZFS 0.7.0

ABD

Slab → memory fragmentation
→ wasted memory

OpenZFS 0.7.0

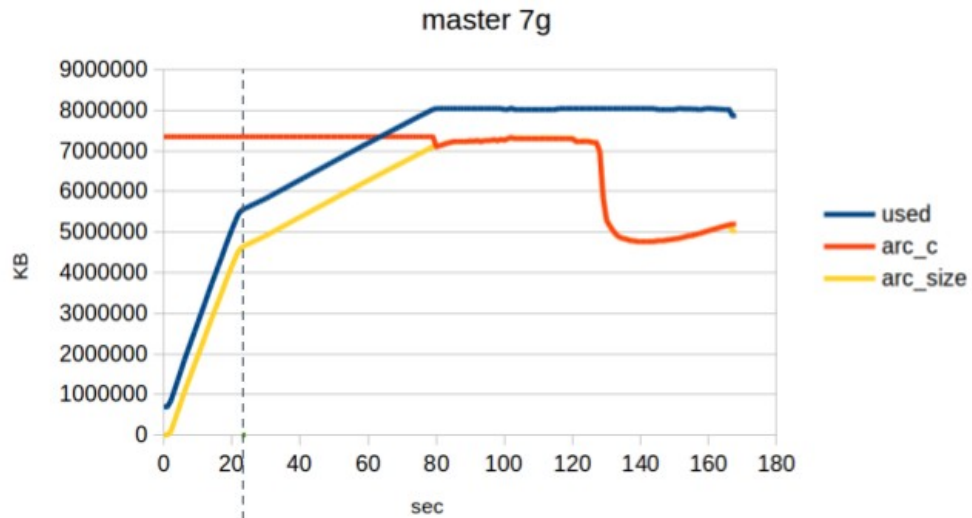
ABD

Slab → memory fragmentation
→ wasted memory

→ user data out of SLAB
scatter/gather

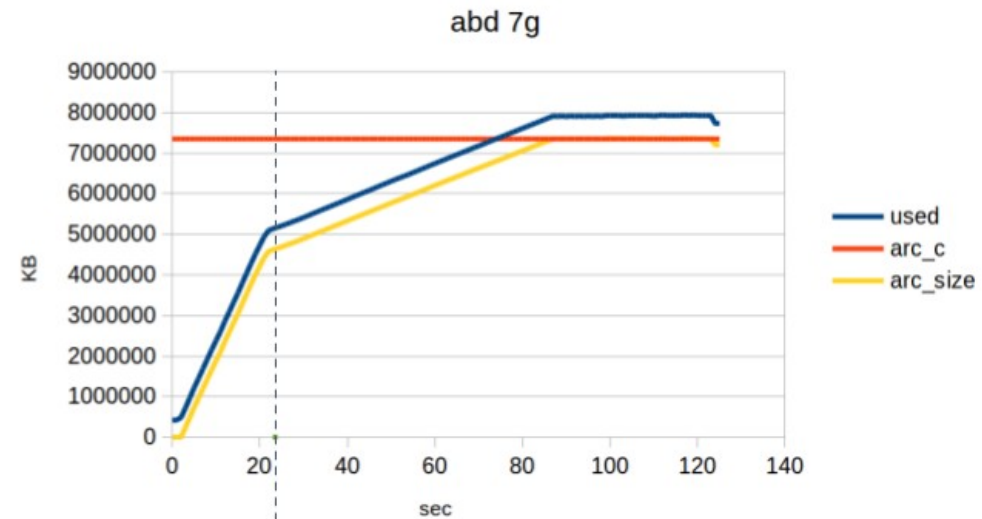
OpenZFS 0.7.0

ABD



file IO end

file: 175906KB/s
zvol: 25509KB/s



file IO end

file: 178405KB/s
zvol: 34565KB/s

OpenZFS 0.7.0

Compressed ARC

HW optimalizace (vektORIZACE raidz, fletcher4)

New checksums: SHA-512, Skein, or Edon-R

dnode accounting

Basic JBOD management

OpenZFS 0.7.0

`zfs send -c`

→ compressed send

`zfs send -t <token> | ssh ... zfs recv -s pool/dset`

→ resumable send/recv

`zpool iostat -w | -l | -r`

→ request histogram | latency | request size

`zpool scrub -s`

→ scrub pause

OpenZFS Encryption

Teď: Nad/pod ZFS (dmccrypt/ecryptfs)

Native ZFS Encryption?

- výkon
- čistější implementace
- snadnější správa
- zálohy nepotřebují klíče

→ block level @ ZFS stack

OpenZFS Encryption

zfs get | set

```
encryption=off \
  | aes-128-ccm \
  | aes-192-ccm \
  | aes-256-ccm | on \
  | aes-128-gcm \
  | aes-192-gcm \
  | aes-256-gcm
```

keyformat=raw | hex | passphrase

pbkdf2iters=350000 (> 100k)

keylocation=prompt | file://



OpenZFS Encryption

zfs get keystatus

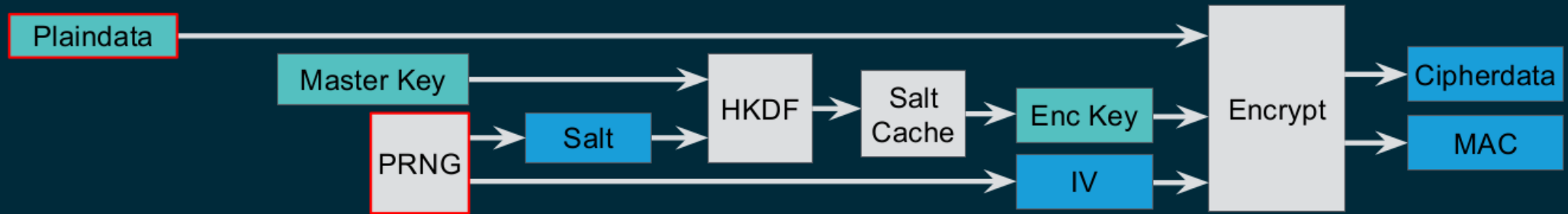
zfs load-key [-nr] [-L keylocation] -a | filesystem

zfs unload-key [-r] -a | filesystem

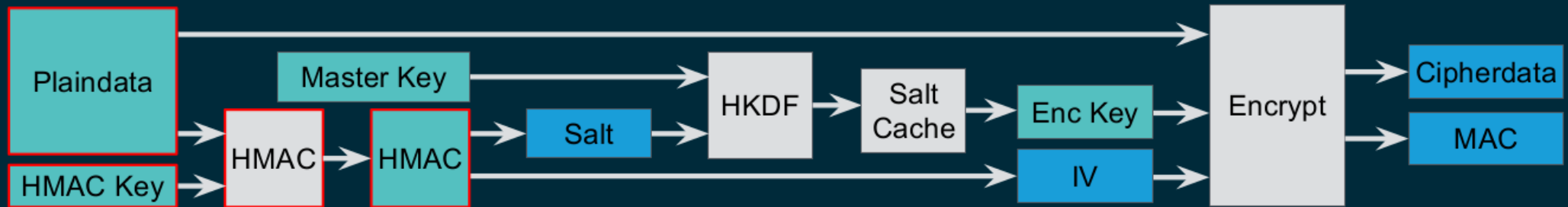
zfs change-key -i [-l] [-o options] filesystem
options: keyformat, keylocation, pbkdf2iters

OpenZFS Encryption

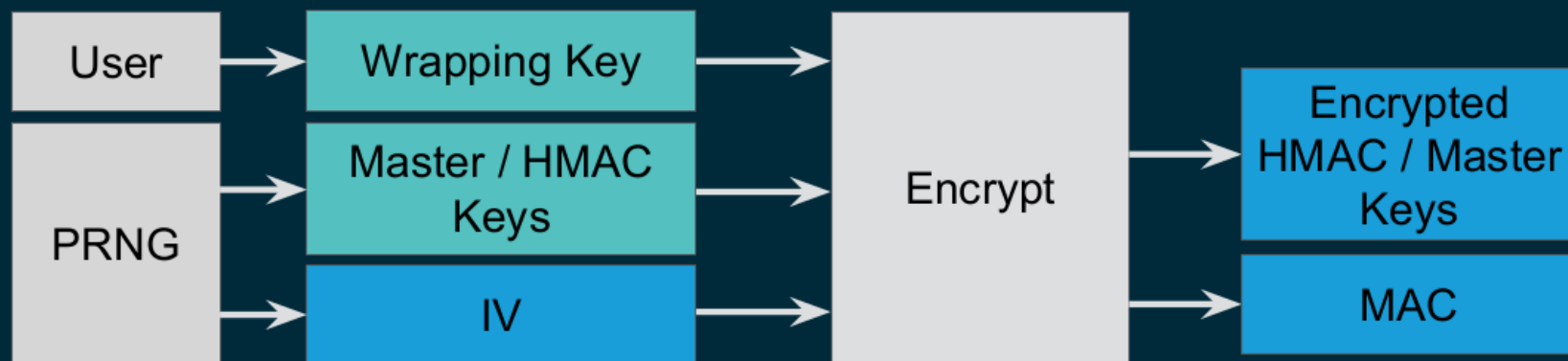
Non-Dedup



Dedup



OpenZFS Encryption



OpenZFS Encryption

Problems

copies=2

dedup leaks, limited to same key

CRIME

→ compression=off

OpenZFS dRAID

Traditional RAIDZ

raidz1-0

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
.

raidz1-1...raidz1-2

.

raidz1-3

15	16	17	18	19
15	16	17	18	19
15	16	17	18	19
15	16	17	18	19
.

raidz1-4

20	21	22	23	24
20	21	22	23	24
20	21	22	23	24
20	21	22	23	24
.

raidz1-5

25	26	27	28	29
25	26	27	28	29
25	26	27	28	29
25	26	27	28	29
.

hot spare

30
30
30
30
.

Declustered RAID

draid1-0

distributed spare

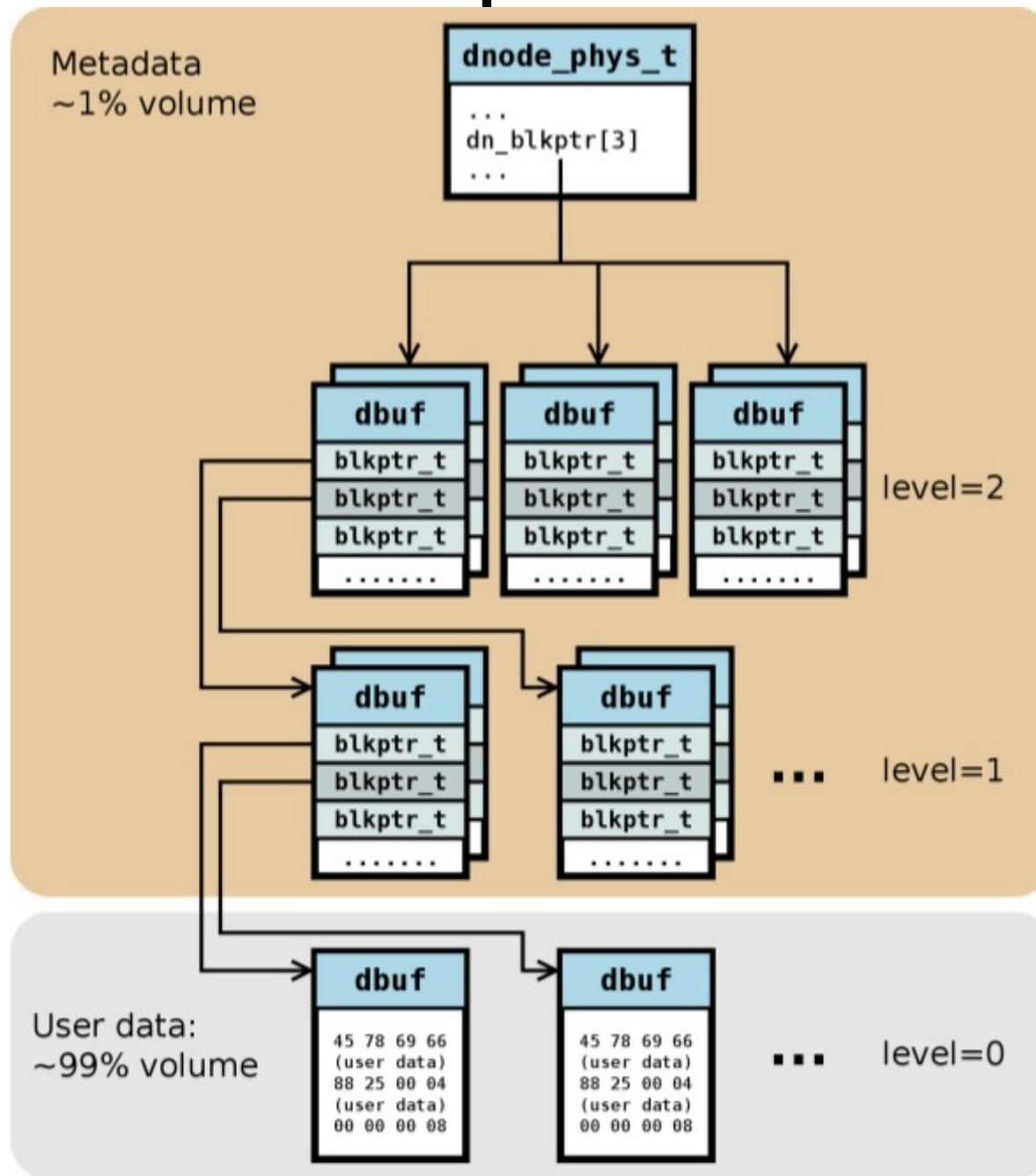
4	8	2	16	1	29	27	30	23	15	28	25	14	19	7	11	22	21	13	26	
5	9	3	17	2	30	28	0	24	16	29	26	15	20	8	12	23	22	14	27	
6	10	4	18	3	0	29	1	25	17	30	27	16	21	9	13	24	23	15	28	
7	11	5	19	4	1	30	2	26	18	0	28	17	22	10	14	25	24	16	29	
.

0
1
2
3
.

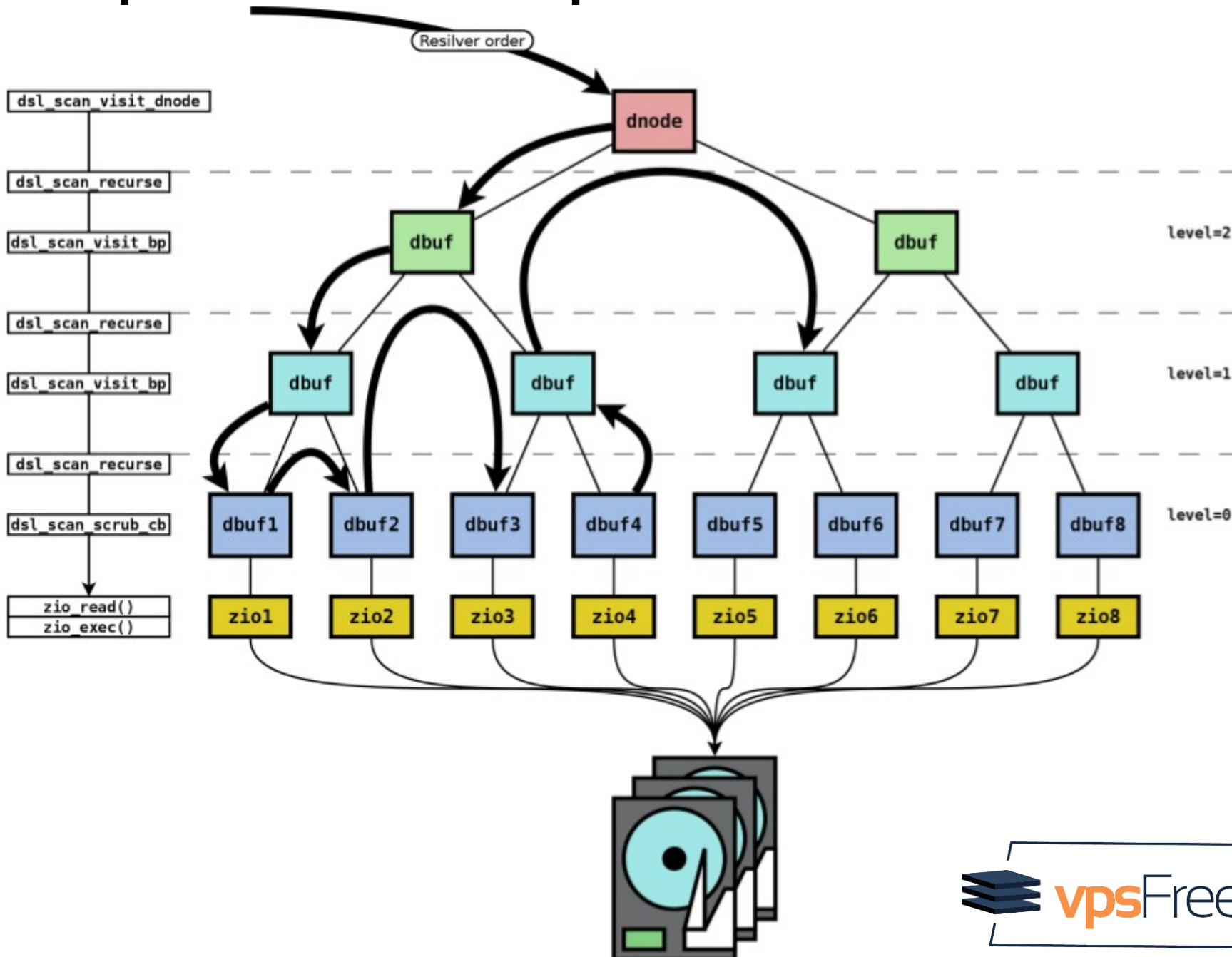
<https://github.com/zfsonlinux/zfs/pull/5841>



OpenZFS Sequential Resilver

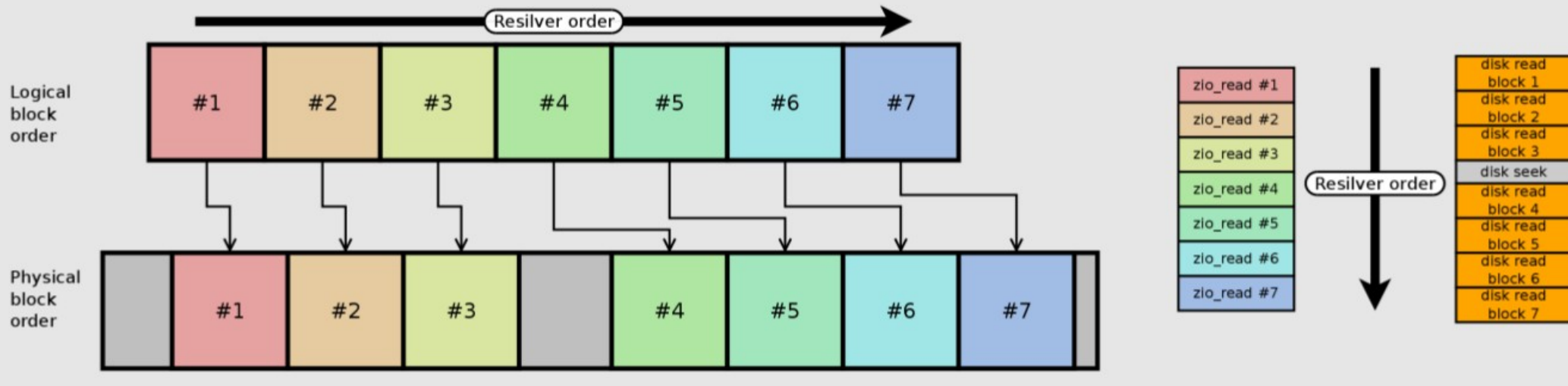


OpenZFS Sequential Resilver

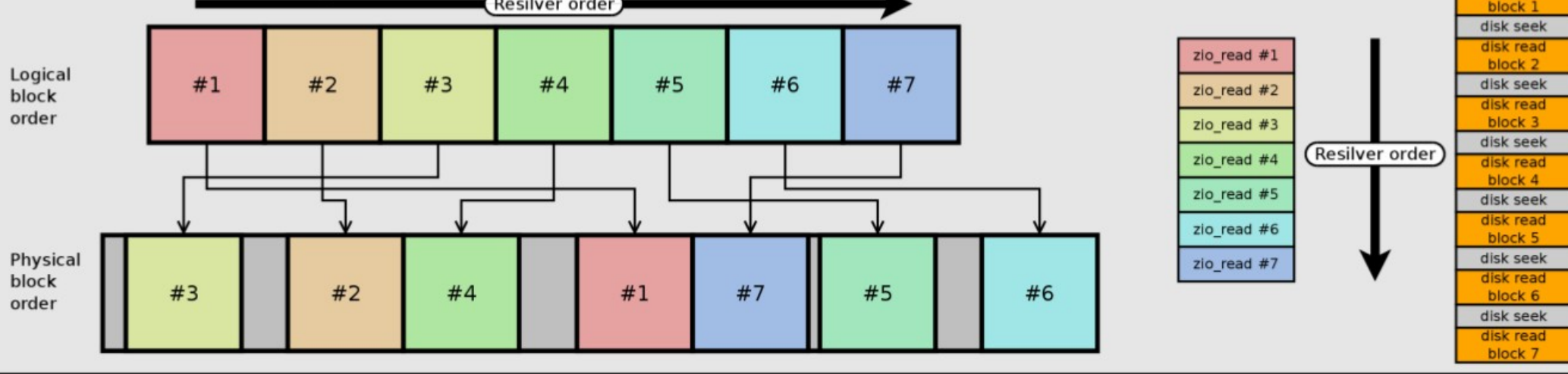


OpenZFS Sequential Resilver

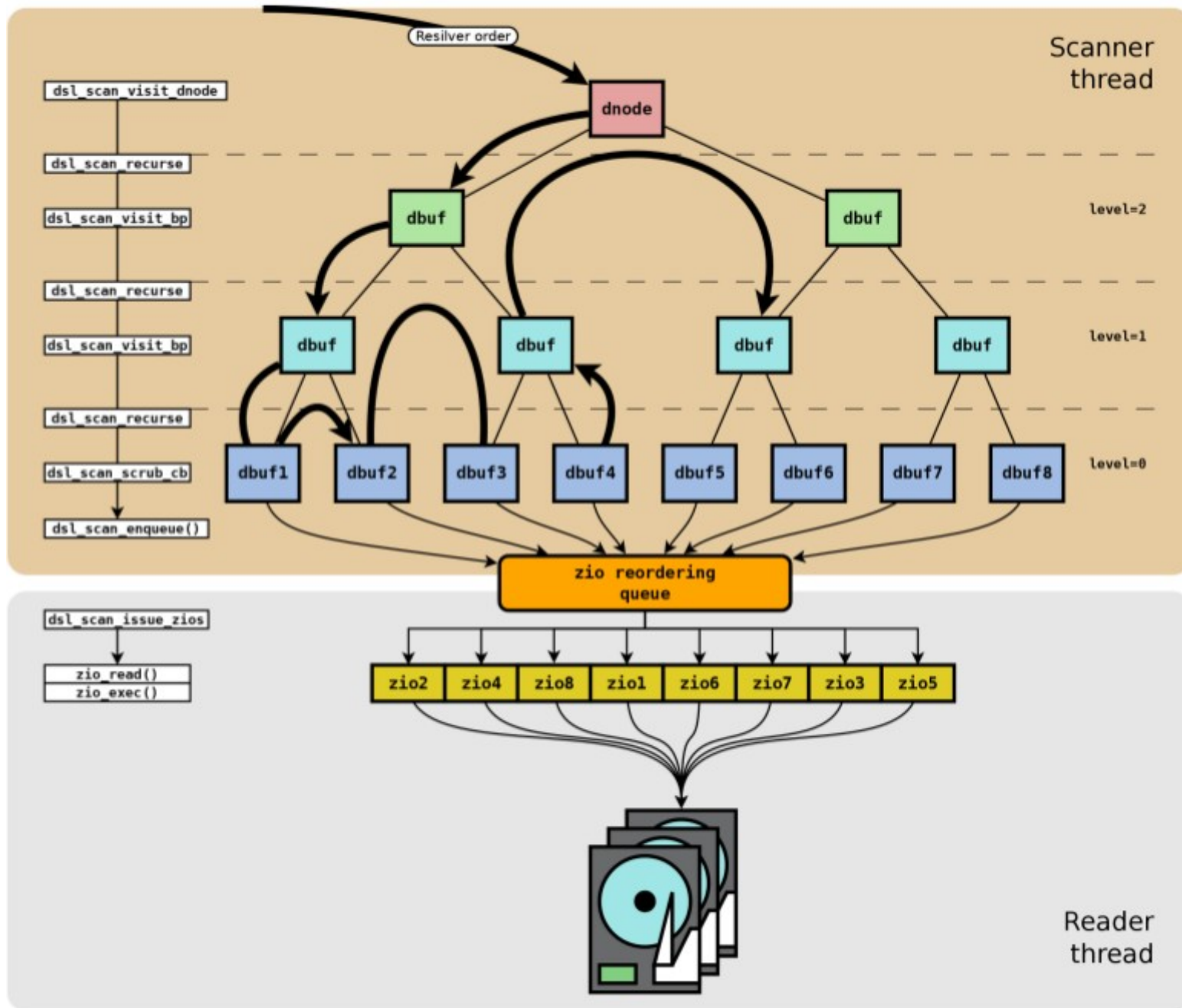
On initial writing



After lots of rewriting



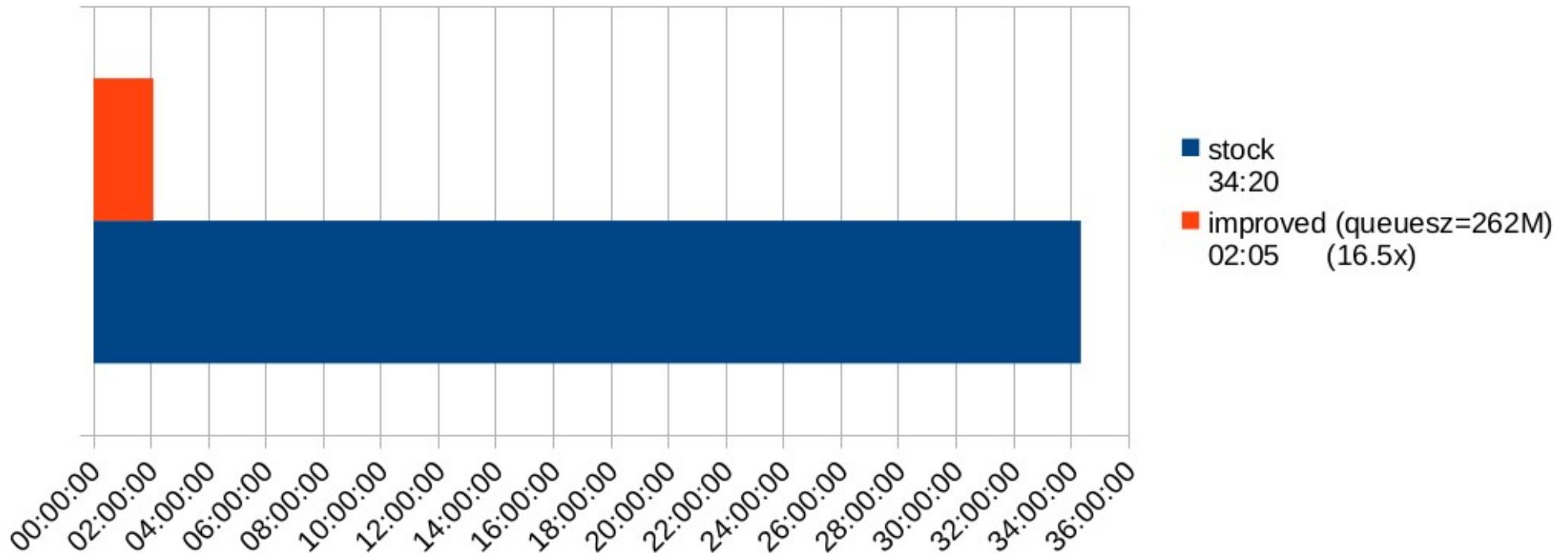
OpenZFS Sequential Resilver



OpenZFS Sequential Resilver

Scrub time (lower is better)

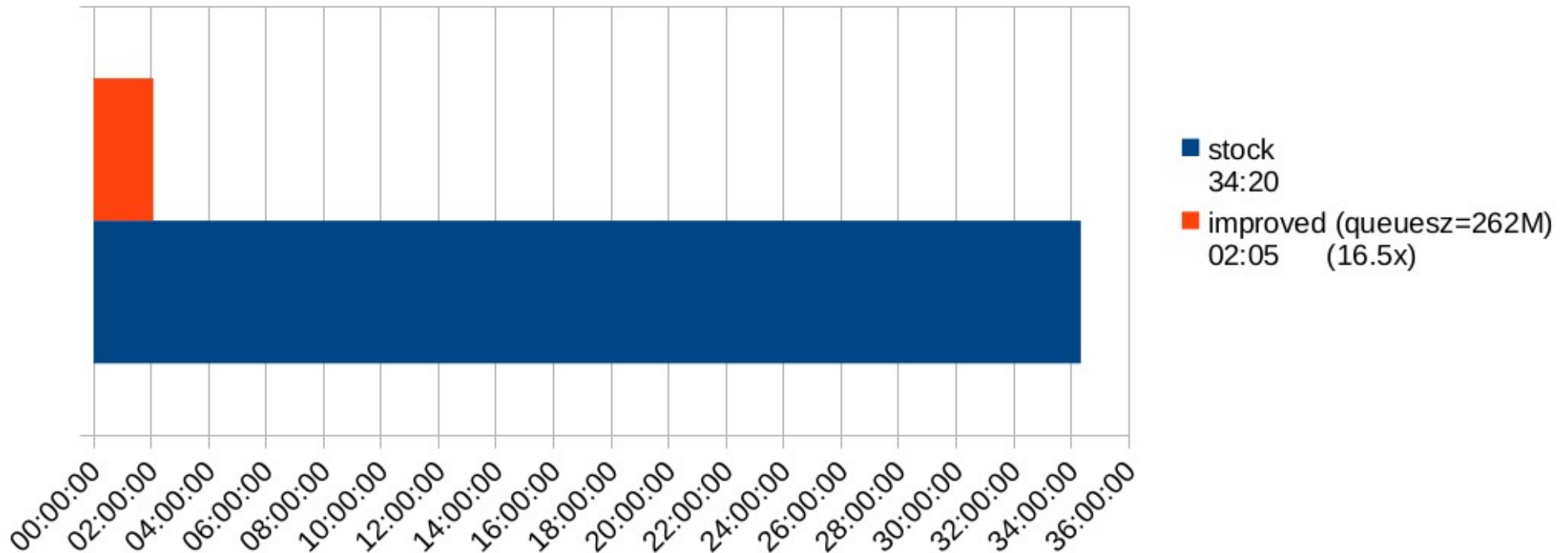
998 GB data (71% cap), 1x5 raidz2 (300GB HDDs)



OpenZFS Sequential Resilver

Scrub time (lower is better)

998 GB data (71% cap), 1x5 raidz2 (300GB HDDs)



<https://github.com/zfsonlinux/zfs/pull/6256>

Device evacuation/removal

zpool remove disk

→ virtual VDEV + indirection table

zfs remap

<https://github.com/openzfs/openzfs/pull/251>

Metadata Allocation Classes

<https://github.com/zfsonlinux/zfs/pull/5182>

```
$ zpool list -o name,segregate_log,segregate_special
NAME          SEGREGATE_LOG  SEGREGATE_SPECIAL
demo          off            on
spill-test    off            off
```

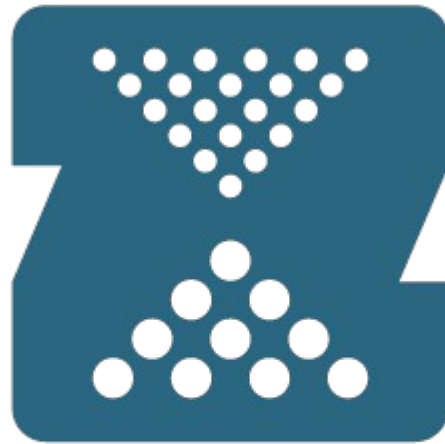
```
[root@ssu1_oss2]# zpool list -C ssu_1ost1
NAME          SIZE      ALLOC    FREE    CAPACITY
-----
ssu_1ost1     72.7T    56.0T    16.8T    77.0%
  draid2-0    72.7T    56.0T    16.8T    77.0%
    normal    58.2T    55.8T    2.48T    95.8%
    special    2.25T    217G    2.04T    9.43%
  unassigned  12.2T      0    12.2T     -
```



OpenZFS Developer Summit

A proposal for 1,000x better dedup performance

RAID-Z Expansion



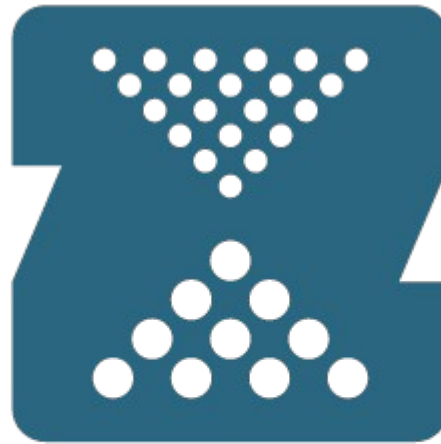
OpenZFS



QA

?

→ snajpa@snajpa.net



OpenZFS

