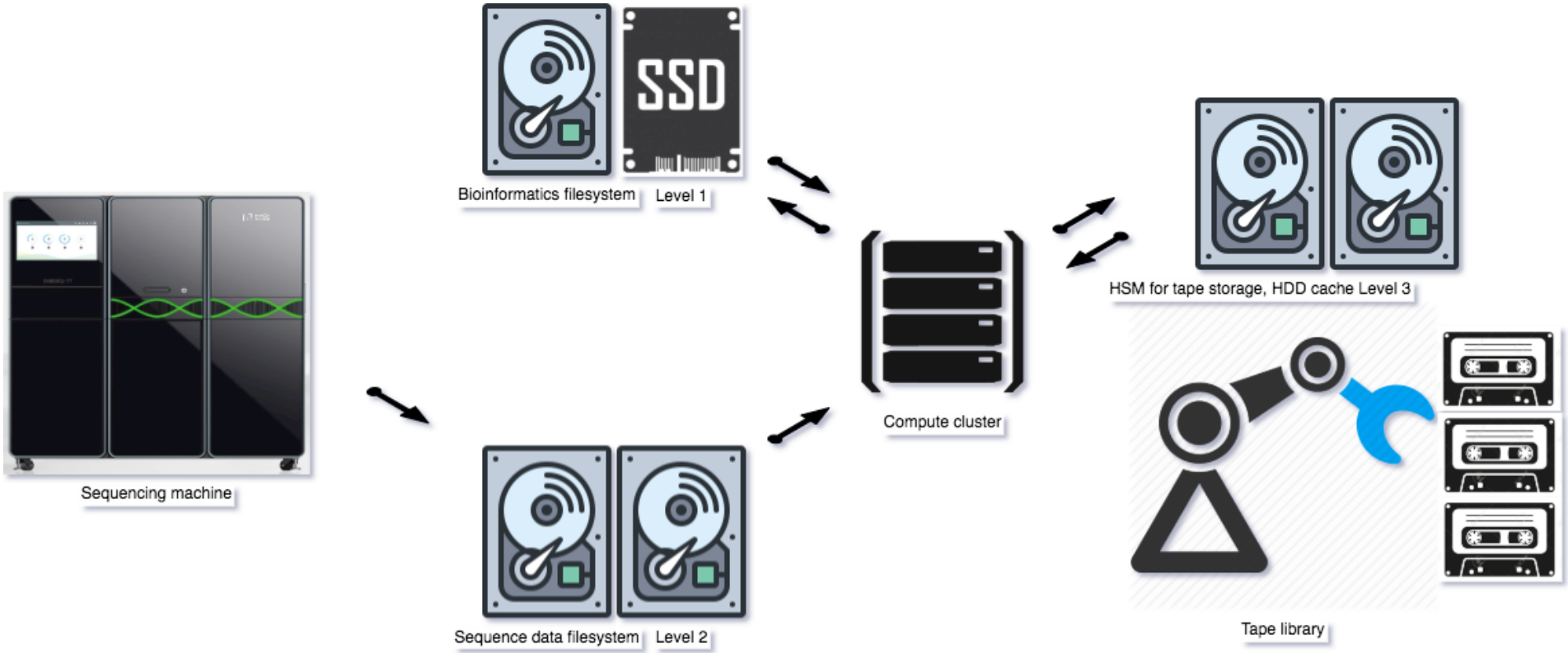
OpenZFS in storage system

China National GeneBank
HPC engineer
Homer Li 李焱
2020-09

Data stream



Near-surface exploration

exploration 1 → exploration 2 → exploration 3

Management

- Different internal and external topology

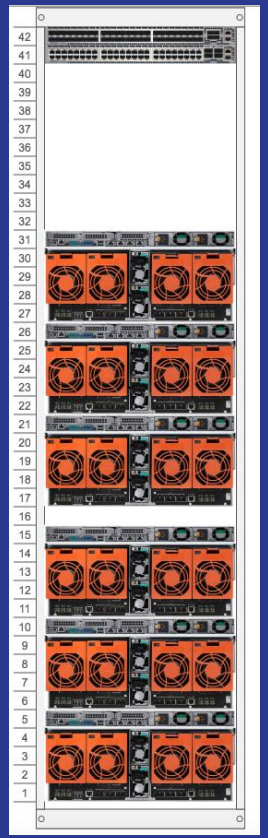
Gaps

- Recordsize
- Ashfit
- Device number

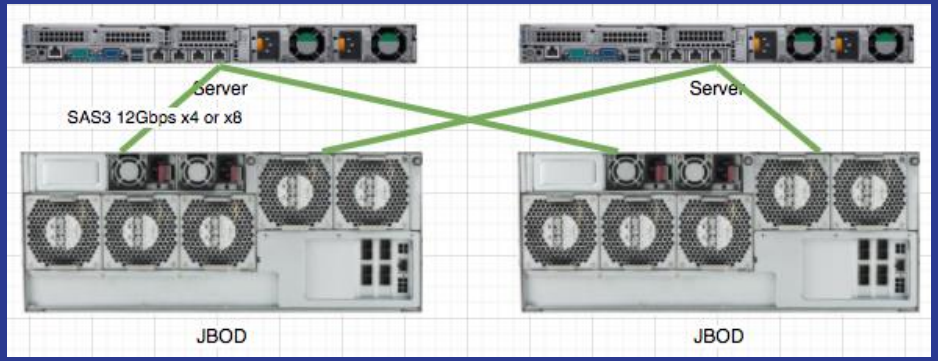
Performance

- Good enough in throughput
- The writing impact the read
- Small block size read when cache miss

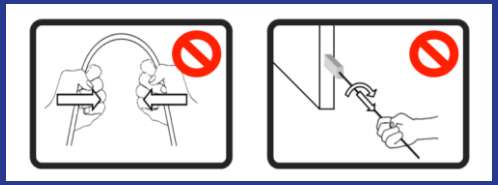
JBOD + ZFS + distributed system(Lustre/others)



- High density JBOD
- LSI 9300-8e 12Gbps SAS HBA
  - SAS bandwidth half duplex (x8 wide bus)
    - Single port 4800 MB/s, Dual port 9600 MB/s
    - The 16 ports or 8 ports adapter only PCIE Gen3 x8, 7876MB/s
  - External Mini SAS HD 4x 12 Gbps Cable (SFF-8644)
    - External receptacles and copper cable assemblies
    - About 3500MB~3756MB/s with an external 12Gbps x 4 lanes cable
    - 20 x NL SAS 2x(8+2 raidz2), READ: 2.3~2.4GB/s, WRITE: 2.8~3GB/s



Be careful Shaolin iron finger when deploy



ZFS issues in the production

Welcome to play WHAC-A-MOLE

Driver name out of order

Zpool status show the failed disk was “sdfl1”
Kernel show the sdfl wwid: “4abb” , and sdfl was online

Attach a disk: sdcw, wwid: “d9e8”

```
# grep -A 2 scsi-35000c500a65b207b zpool_status
            scsi-35000c500a65b207b  ONLINE       0     0     0
            14718944552882581455    FAULTED      0     0     0  was /dev/sdfl1
            scsi-35000c500a670cbc7  ONLINE       0     0     0
# grep sdfl lsscsi.log
[6:0:176:0]  disk    SEAGATE  ST10000NM0256    TT54  /dev/sdfl  35000c500a6754abb  /dev/sg176  9.79TB
# grep 35000c500a6754abb zpool_status
            scsi-35000c500a6754abb  ONLINE       0     0     0
# grep 35000cca273a7d9e8 lsscsi.log
[6:0:106:0]  disk    HGST     HUH721010AL5200  LS15  /dev/sdcw  35000cca273a7d9e8  /dev/sg106  9.79TB
```

Same WWID

```
children[0]:
    type: 'disk'
    id: 0
    guid: 14718944552882581455
    path: '/dev/sdfl1'
    devid: 'scsi-35000cca273a7d9e8-part1'
    phys_path: 'pci-0000:b3:00.0-sas-0x5000cca273a7d9e9-lun-0'
    vdev_enc_sysfs_path: '/sys/class/enclosure/17:0:88:0/68'
    whole_disk: 1
    not_present: 1
    DTL: 149
    create_txg: 4
    com.delphix:vdev_zap_leaf: 88
children[1]:
    type: 'disk'
    id: 1
    guid: 7602460441426092393
    path: '/dev/sdcw1'
    devid: 'scsi-35000cca273a7d9e8-part1'
    phys_path: 'pci-0000:b3:00.0-sas-0x5000cca273a7d9e9-lun-0'
    vdev_enc_sysfs_path: '/sys/class/enclosure/6:0:89:0/68'
    whole_disk: 1
    DTL: 182
    create_txg: 4
    com.delphix:vdev_zap_leaf: 403
    resilver_txg: 49772215
```

```
# zpool replace ost_0 14718944552882581455 /dev/sdcw
invalid vdev specification
use '-f' to override the following errors:
/dev/sdcw1 is part of active pool 'ost_0'
# zpool replace -f ost_0 14718944552882581455 /dev/sdcw
invalid vdev specification
the following errors must be manually repaired:
/dev/sdcw1 is part of active pool 'ost_0'
```

Slow device

```
    inquiry cdb: 12 01 00 00 fc 00
    inquiry: pass-through requested 252 bytes but got 23 bytes
    inquiry cdb: 12 01 80 00 fc 00
standard INQUIRY:
  PQual=0  Device_type=0  RMB=0  version=0x06  [SPC-4]
  [AERC=0]  [TrmTsk=0]  NormACA=0  HiSUP=1  Resp_data_format=2
  SCCS=0  ACC=0  TPGS=0  3PC=0  Protect=1  [BQue=0]
  EncServ=0  MultiP=1 (VS=1)  [MChngr=0]  [ACKREQQ=0]  Addr16=0
  [RelAdr=0]  WBus16=0  Sync=0  Linked=0  [TranDis=0]  CmdQue=1
  [SPI: Clocking=0x0  QAS=0  IUS=0]
    length=144 (0x90)   Peripheral device type: disk
 Vendor identification:
 Product identification:
 Product revision level:
 Unit serial number:
    inquiry: pass-through requested 252 bytes but got 12 bytes

real    1m2.095s
user    0m0.000s
sys     0m0.001s
```

- After reboot, if the single device does not respond in the time, it will cause zpool to suspend again
- Responds after a few minutes

```
sd 0:0:160:0: attempting task abort! scmd(ffff8e24a2179880)
sd 0:0:160:0: [sdev] tag#0 CDB: Read(32)
sd 0:0:160:0: [sdev] tag#0 CDB[00]: 7f 00 00 00 00 00 00 18 00 09 20 00 00 00 00 00
sd 0:0:160:0: [sdev] tag#0 CDB[10]: e7 23 01 48 e7 23 01 48 00 00 00 00 00 00 00 89
scsi target0:0:160: _scsih_tm_display_info: handle(0x00b3), sas_address(0x5000cca2736aaadd), phy(19)
scsi target0:0:160: enclosurelogical id(0x50050cc11ac016e2), slot(30)
scsi target0:0:160: enclosure level(0x0000), connector name( 1   )
sd 0:0:160:0: task abort: SUCCESS scmd(ffff8e24a2179880)
sd 0:0:160:0: [sdev] tag#0 FAILED Result: hostbyte=DID_TIME_OUT driverbyte=DRIVER_OK
sd 0:0:160:0: [sdev] tag#0 CDB: Read(32)
sd 0:0:160:0: [sdev] tag#0 CDB[00]: 7f 00 00 00 00 00 00 18 00 09 20 00 00 00 00 00
sd 0:0:160:0: [sdev] tag#0 CDB[10]: e7 23 01 48 e7 23 01 48 00 00 00 00 00 00 00 89
blk_update_request: I/O error, dev sdev, sector 3877830984
sd 0:0:160:0: attempting task abort! scmd(ffff8e24a217b9c0)
sd 0:0:160:0: [sdev] tag#4 CDB: Read(32)
sd 0:0:160:0: [sdev] tag#4 CDB[00]: 7f 00 00 00 00 00 00 18 00 09 20 00 00 00 00 00
sd 0:0:160:0: [sdev] tag#4 CDB[10]: e7 23 00 36 e7 23 00 36 00 00 00 00 00 00 00 89
```

```
WARNING: MMP writes to pool 'ost_100' have not succeeded in over 159s; suspending pool
WARNING: Pool 'ost_100' has encountered an uncorrectable I/O failure and has been suspended.
```

Trace the slow device

```
================== All Devices ==================

            ALL           MIN           AVG           MAX           N
--------------- ------------- ------------- ------------- -----------

Q2Q               0.000000001   0.003806706   1.198522962      472848
Q2G               0.000000514   0.000588118   0.253734427      349171
S2G               0.001479053   0.117520681   0.253732342        1741
G2I               0.000000500   0.000028340   0.009261576      349171
Q2M               0.000000263   0.000000558   0.000077001      123678
I2D               0.000000503   0.000010245   0.000148244      349171
M2D               0.000002922   0.000133254   0.009259388      123678
D2C               0.000137234   0.234211851   2.667617720      472837
Q2C               0.000142403   0.234709646   2.667628087      472837

================== Device Overhead ==================

       DEV |       Q2G       G2I       Q2M       I2D       D2C
---------- | --------- --------- --------- --------- ---------
 (  8,160) |   0.1850%   0.0089%   0.0001%   0.0032%  99.7879%
---------- | --------- --------- --------- --------- ---------
   Overall |   0.1850%   0.0089%   0.0001%   0.0032%  99.7879%

================== Device Merge Information ==================

       DEV |       #Q       #D   Ratio |   BLKmin   BLKavg   BLKmax    Total
---------- | -------- -------- ------- | -------- -------- -------- --------
 (  8,160) |   472849   349171     1.4 |        8       12       24  4473520
```

- Only half throughput
- The fluctuation of latency over time
- Smart health is OK

- A branch of devices IO error
- HBA command timeout

```
sd 14:0:184:0: device_unblock and setting to running, handle(0x0064)
sd 14:0:185:0: device_unblock and setting to running, handle(0x0065)
sd 14:0:186:0: device_unblock and setting to running, handle(0x0066)
sd 14:0:187:0: device_unblock and setting to running, handle(0x0067)
sd 14:0:101:0: rejecting I/O to offline device
sd 14:0:101:0: [sdij] killing request
sd 14:0:101:0: rejecting I/O to offline device
sd 14:0:101:0: [sdij] FAILED Result: hostbyte=DID_NO_CONNECT driverbyte=DRIVER_OK
sd 14:0:101:0: [sdij] CDB: Read(16) 88 00 00 00 00 00 c9 96 68 77 00 00 00 01 00 00
blk_update_request: I/O error, dev sdij, sector 3382077559
sd 14:0:101:0: [sdij] killing request
sd 14:0:101:0: [sdij] FAILED Result: hostbyte=DID_NO_CONNECT driverbyte=DRIVER_OK
sd 14:0:101:0: [sdij] CDB: Write(16) 8a 00 00 00 00 04 8c 3f b5 fe 00 00 00 02 00 00
blk_update_request: I/O error, dev sdij, sector 19532854782
sd 14:0:101:0: [sdij] Synchronizing SCSI cache
sd 14:0:101:0: [sdij] Synchronize Cache(10) failed: Result: hostbyte=DID_NO_CONNECT driverbyte=DRIVER_OK
```

```
mpt3sas_cm0: Command Timeout
 mf:
        00000011 00000000 00000000 00000000 00000000 00000018 00000000 0000012c
        00000000 00000006 00000000 00000000 00000000 00000000 00000000 02000000
        00000012 0000002c 00000000 00000000 00000000 00000000 00000000 00000000
mpt3sas_cm0: issue target reset: handle = (0x0011)
```

Slow device question

The issuing vendor reply:

The drives show an increasing value in log page 03h, parameter code 0000 (ECC on-the-fly counter).

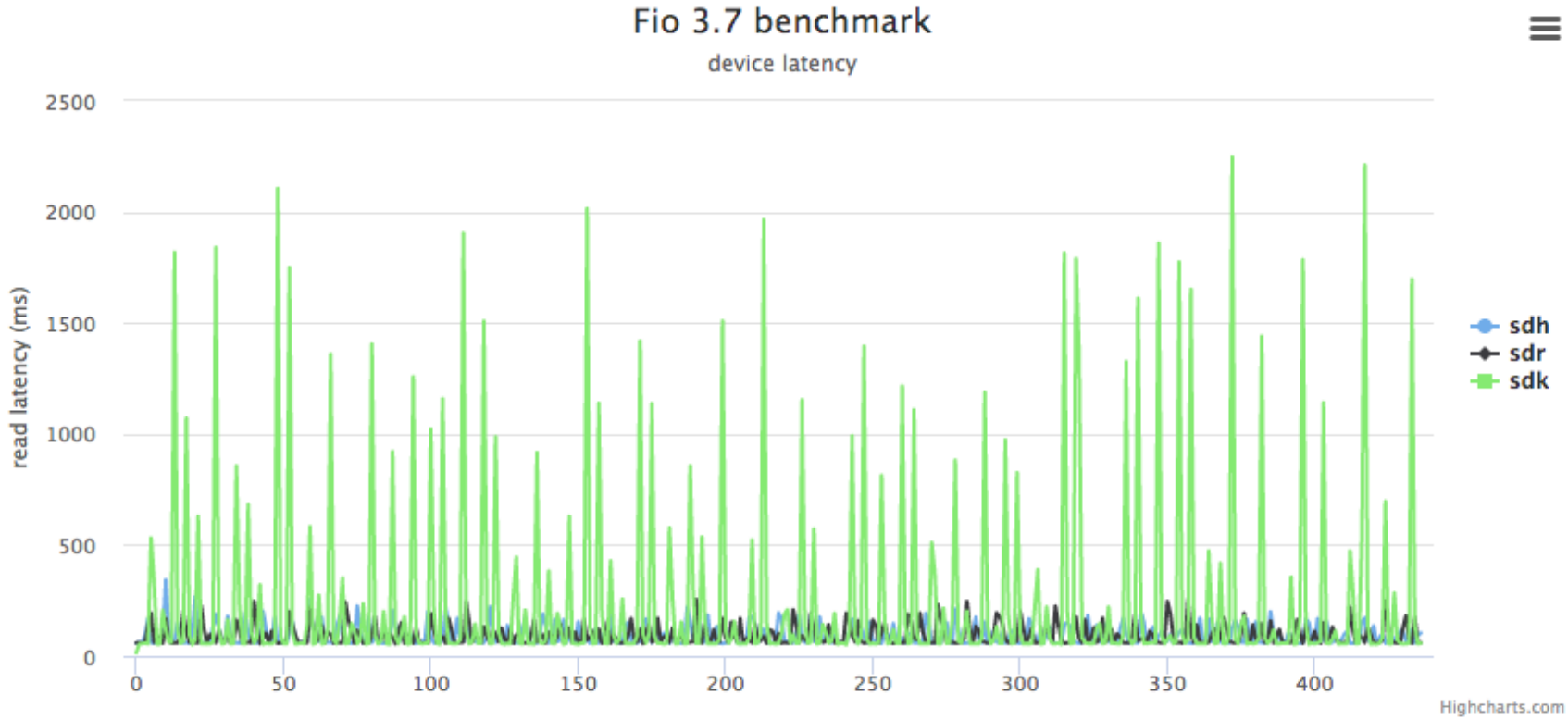
ECC on-the-fly is necessary for HDDs to function properly at current areal densities. Rate of ECC on the fly can vary based on drive model, drive capacity, areal density, disk speed, and environmental factors.

No risk during standard operation. The counter will increase as part of normal operation

1. The Competitors support this parameters
2. Just read error, no others
3. Competitor value was extremely slower than the issuing vendor
4. Impact the performance

NL-SAS HDD in the same JBOD	Vendor (err/corrected)	Issuing vendor
Write error	0/0	76/76
Read error	0/ 2170	0/1100081991
Verify error	0/1	0/24415719
Non medium	0	17183
Read TB	472.75	591.55
Write TB	34.12	35.21

Slow device latency



Performance fluctuation

- High loading
  - Get high latency device by log
  - Failed disappeared after high loading
  - Offline and Self test
    - Show error
      - replace
    - Pass test ?
      - Device overloading

```
=== START OF READ SMART DATA SECTION ===
SMART Health Status: FAILURE PREDICTION THRESHOLD EXCEEDED: ascq=0xfd [asc=5d, ascq=fd]

Current Drive Temperature:     31 C
Drive Trip Temperature:        85 C
```

```
=== START OF READ SMART DATA SECTION ===
SMART Health Status: OK

Current Drive Temperature:     31 C
Drive Trip Temperature:        85 C
```

```
  await r_await w_await  svctm  %util
  4.12    4.77    4.00    0.24   7.90
2699.92 2577.08 2745.69  14.29 100.00
```

```
  await r_await w_await  svctm  %util
  2.79    9.46    0.42    0.31   6.10
2547.94 2158.22 2685.28  14.08 100.00
```

```
  await r_await w_await  svctm  %util
  0.80    1.64    0.56    0.15   4.80
2248.82 1335.71 2368.29  16.53 100.00
```

```
Num  Test              Status                  segment  LifeTime  LBA_first_err [SK ASC ASQ]
     Description                               number   (hours)
# 1  Background short  Failed in segment -->       3   29064                 - [-   -    -]
```

Import failed 😓

Do not use -F parameter, if you want , use -F and -n

In this cause, it ‘s show the zpool could not be imported

```
$ zpool import -f -o  cachefile=none -o readonly=on ost_61
cannot import 'ost_61': I/O error Destroy and re-create the pool from a backup source.

$ zpool import show

     pool: ost_61
     id: 15498746923132605816
  state: FAULTED
 status: The pool metadata is corrupted.
 action: The pool cannot be imported due to damaged devices or data.
        The pool may be active on another system, but can be imported using
        the '-f' flag.
   see: http://zfsonlinux.org/msg/ZFS-8000-72
 config:        ost_61                      FAULTED  corrupted data
          raidz3-0                  ONLINE
            sddd                    ONLINE
            sdde                    ONLINE
            sddf                    ONLINE
            sddg                    ONLINE
            sddh                    ONLINE
            sddi                    ONLINE
            scsi-35000c500a66c86f7  ONLINE
            scsi-35000c500a675b907  ONLINE
            scsi-35000c500a665afe7  ONLINE
            scsi-35000c500a670c7a3  ONLINE
            scsi-35000c500a66d665f  ONLINE
            scsi-35000c500a65a86b7  ONLINE
            scsi-35000c500a664a617  ONLINE
```

Import failed by single device

Got the mess timestamp and wrong zpool info in sdi

```
children[20]:
                    type: 'disk'
                    id: 20
                    guid: 3278057294213455851
                    whole_disk: 1
                    DTL: 257
                    create_txg: 4
                    path: '/dev/disk/by-id/scsi-35000c500a665b1b3-part1'
                    devid: 'scsi-35000c500a665b1b3-part1'
                    phys_path: 'pci-0000:b3:00.0-sas-0x5000c500a665b1b1-lun-0'
                children[21]:
                    type: 'disk'
                    id: 21
                    guid: 10749955888611927037
                    whole_disk: 1
                    DTL: 256
                    create_txg: 4
                    path: '/dev/disk/by-id/scsi-35000c500a670c5fb-part1'
                    devid: 'scsi-35000c500a670c5fb-part1'
                    phys_path: 'pci-0000:b3:00.0-sas-0x5000c500a670c5f9-lun-0'
        rewind-policy:
            rewind-request-txg: 18446744073709551615
            rewind-request: 2
zdb: can't open 'ost_61': Input/output error
```

```
$ zdb -e ost_61 -d scsi-35000c500a665afe7
Dataset mos [META], ID 0, cr_txg 4, 509M, 284 objects

    Object  lvl   iblk   dblk  dsize  dnsize  lsize   %full  type
         0    2   128K    16K   736K     512   592K   23.99  DMU dnode

$ zdb -e ost_61 -d sddi
zdb: can't open 'ost_61': Input/output error
```

```
echo 1 > /sys/class/block/sddi/device/delete
$ zpool import -f -o  cachefile=none -o readonly=on ost_61
$ echo $?
0
$ zpool status -x
  pool: ost_61
 state: DEGRADED
status: One or more devices could not be used because the label is missing or
        invalid.  Sufficient replicas exist for the pool to continue
        functioning in a degraded state.
action: Replace the device using 'zpool replace'.
   see: http://zfsonlinux.org/msg/ZFS-8000-4J
  scan: resilvered 4.59G in 0h8m with 0 errors on Tue Dec  4 14:12:32 2018
config:

        NAME                      STATE     READ WRITE CKSUM
        ost_61                    DEGRADED     0     0     0
          raidz3-0                DEGRADED     0     0     0
            sddd                  ONLINE       0     0     0
```

Manual rollback by “-T”

behlendorf commented on Oct 30, 2014 — Contributor

@simonbuehler You'll need to make this one line change and rebuild the module. After which you'll be able to use the `-T` option. This effectively disables the logic which prevents you from using uberblocks which are older than the label.

It would still be a good idea to import the pool read-only the first time. If everything appears to go smoothly you can import it read-write. When the new labels are written out out you'll be able to import the without this patch. Let us know how it goes.

```
diff --git a/module/zfs/vdev_label.c b/module/zfs/vdev_label.c
index 1c2f00f..509e812 100644
--- a/module/zfs/vdev_label.c
+++ b/module/zfs/vdev_label.c
@@ -471,7 +471,7 @@ retry:
                        if ((error || label_txg == 0) && !config) {
                                config = label;
                                break;
-                       } else if (label_txg <= txg && label_txg > best_txg) {
+                       } else if (label_txg > best_txg) {
                                best_txg = label_txg;
                                nvlist_free(config);
                                config = fnvlist_dup(label);
```

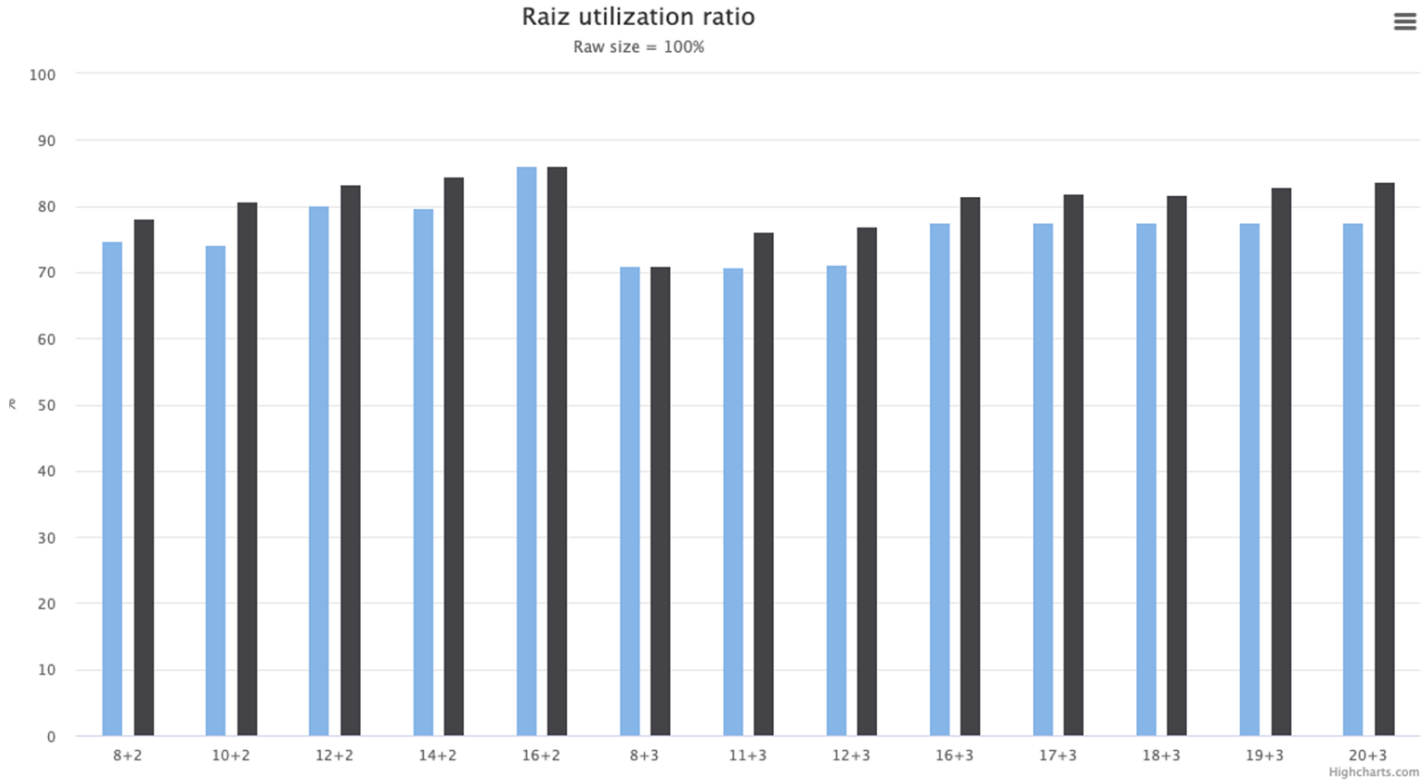
ZFS Overhead and performance

You could get many issues, because zfs is open source and easy to test

RaidZ capacity

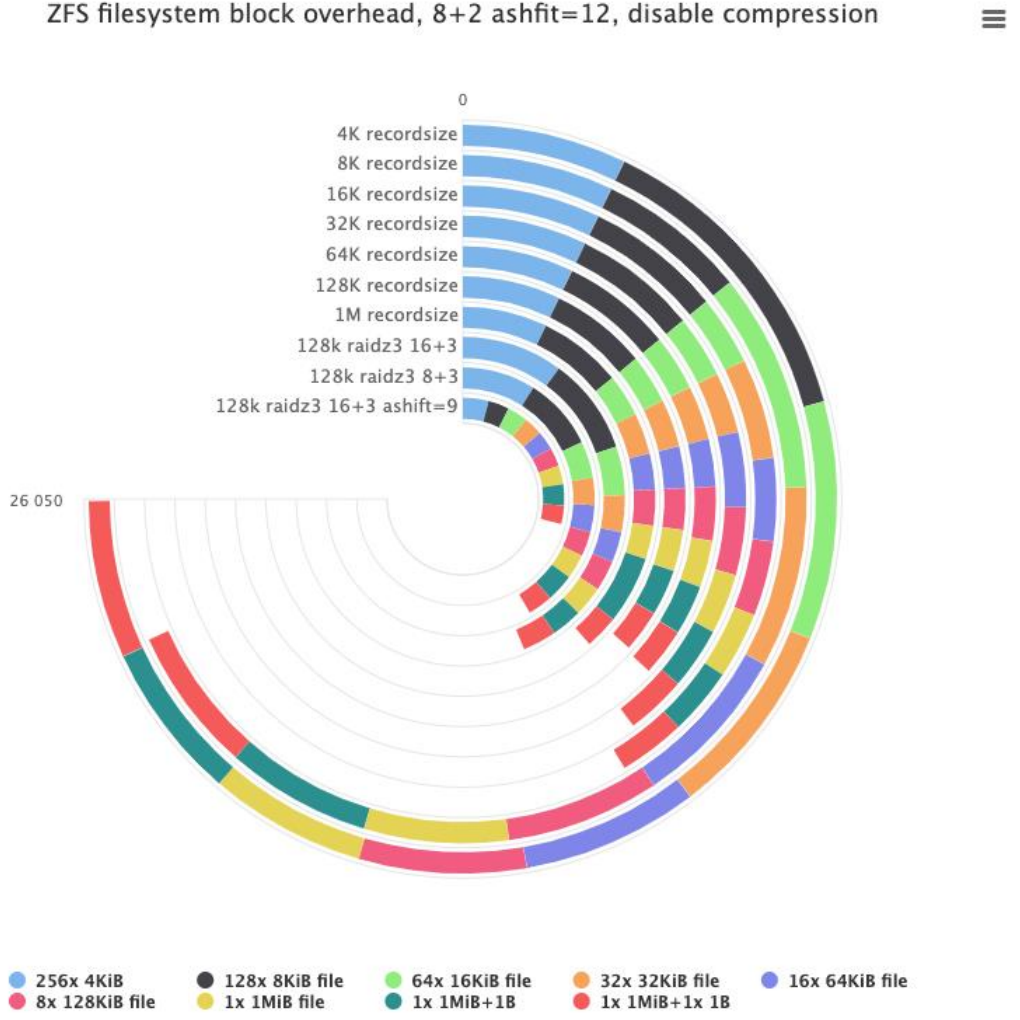
ashfit=12 (blue)
For performance

ashfit=9 (black)
For capacity



Data occupancy cost

- After the zpool created
- The test case, 9x 1MiB files save in different recordsize
- If each cycle has the same angle that means they take the same space
- Eg: The outermost cycle
- 9216KiB files consumed 26050 KiB with 4KiB recordsize
- The innermost cycle takes the least space
- The lower recordsize will take the more extra available space for the same files



Small read block when cache miss

Recordszie=1M	Read 1MiB file
Read 1MiB File by Block size 4K	256MiB/s to HDD
8KiB	128MiB/s
16KiB	64MiB/s
32KiB	32MiB/s
64KiB	16MiB/s
128KiB	8MiB/s
1MiB	1MiB/s

- Ashfit = 12, Raidz2 8+2
- ZFS 0.7.13 and 0.8.4 Posix layer
- ARC miss read

- Set the small read block size could trigger the read amplification
- 64K/128K is a balanced choice, the lower amplification if the ARC cache miss

That why we can ‘t use 1M recordsize for too many tiny files

Add more ARC and L2ARC

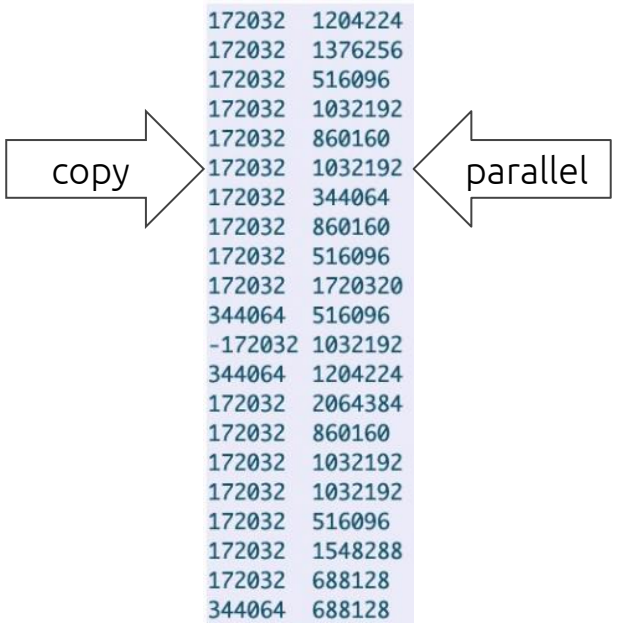
Parallel write impact (Only 0.6 is OK)

CentOS 7 raidz2 8+2, 128K	parallel write dir(read secs)	The same files after cp(secs)
0.8.4(after create)	84~99	60~61
0.8.4(FRAG 13%)	113~119	67~68
0.7.13(after create)	94~100	64~67
0.7.13(FRAG 51%)	133~141	75~77
0.6.5.11(after create)	99~100	101~103

```
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sdb               0.00     0.00 1602.50    0.00    36.08     0.00    46.11     2.98    1.86    1.86    0.00   0.62 100.00
sdc               0.00     0.00 1570.00    0.00    36.68     0.00    47.85     2.97    1.89    1.89    0.00   0.64 100.00
sdd               0.00     0.00 1554.50    0.00    36.13     0.00    47.60     2.98    1.92    1.92    0.00   0.64 100.00
sde               0.00     0.00 1600.00    0.00    36.41     0.00    46.60     2.96    1.85    1.85    0.00   0.62  99.90
sdf               0.00     0.00 1574.00    0.00    35.03     0.00    45.58     2.98    1.89    1.89    0.00   0.64 100.00
sdg               0.00     0.00 1587.50    0.00    35.16     0.00    45.36     2.98    1.88    1.88    0.00   0.63 100.00
sdi               0.00     0.00 1548.00    0.00    36.21     0.00    47.91     2.98    1.92    1.92    0.00   0.65 100.00
sdj               0.00     0.00 1581.00    0.00    35.87     0.00    46.47     2.97    1.90    1.90    0.00   0.63 100.00
sdk               0.00     0.00 1572.50    0.00    34.79     0.00    45.31     2.97    1.89    1.89    0.00   0.64 100.00
sdh               0.00     0.00 1628.00    0.00    37.09     0.00    46.66     2.97    1.82    1.82    0.00   0.61  99.90
```

```
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sdb               0.00     0.00 1006.00    0.00    94.53     0.00   192.43     2.87    2.86    2.86    0.00   0.96  97.00
sdc               0.00     0.00 1094.00    0.00    93.35     0.00   174.75     2.69    2.47    2.47    0.00   0.84  91.90
sdd               0.00     0.00 1042.00    0.00    93.52     0.00   183.81     2.83    2.72    2.72    0.00   0.92  96.15
sde               0.00     0.00  998.00    0.00    94.03     0.00   192.95     2.87    2.88    2.88    0.00   0.97  97.05
sdf               0.00     0.00 1011.00    0.00    94.51     0.00   191.45     2.89    2.87    2.87    0.00   0.97  97.85
sdg               0.00     0.00 1001.00    0.00    92.87     0.00   190.00     2.85    2.83    2.83    0.00   0.96  96.30
sdi               0.00     0.00 1033.50    0.00    93.28     0.00   184.85     2.67    2.58    2.58    0.00   0.88  90.65
sdj               0.00     0.00 1033.00    0.00    92.61     0.00   183.61     2.76    2.68    2.68    0.00   0.90  93.40
sdk               0.00     0.00  992.00    0.00    94.32     0.00   194.73     2.88    2.90    2.90    0.00   0.98  97.20
sdh               0.00     0.00 1197.50    0.00    93.21     0.00   159.42     2.66    2.23    2.23    0.00   0.76  91.10
```

Single process write compare with multiple processes write at the same time, multiple write cause the bad read performance
ZFS blocks allocation
More iops and the lower BW



Release the write throttle in default config

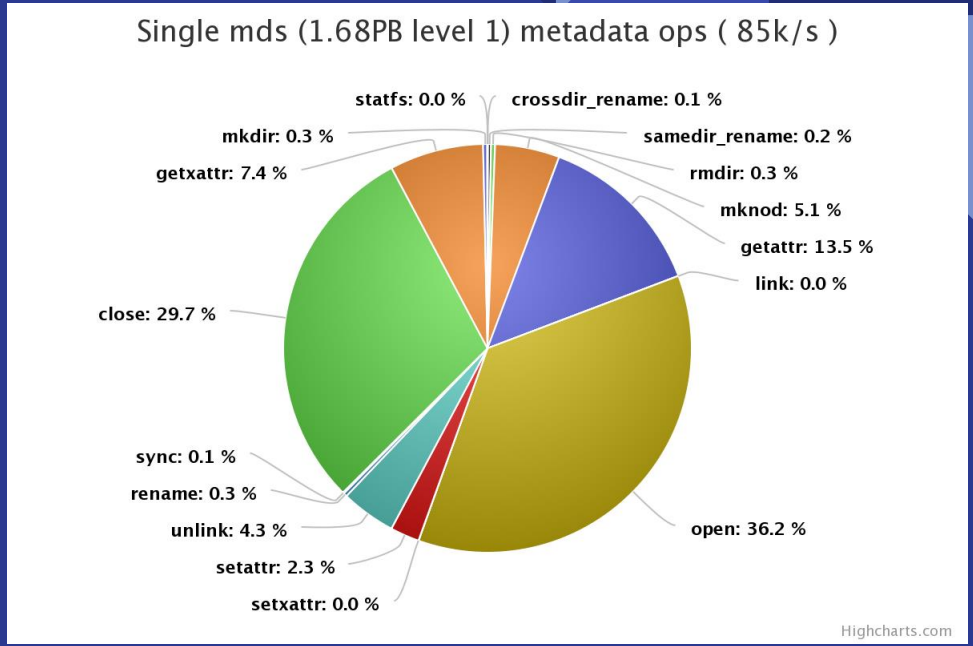
```
$ dd if=/dev/zero of=test bs=1M &
[1] 2945
$ zpool iostat 2
              capacity     operations     bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
tank        85.2M  1.81T      4    167   158K  11.7M
tank        1.09G  1.81T      0  6.06K      0   407M
tank        1.09G  1.81T      0  6.32K      0   436M
tank        1.09G  1.81T      0  6.29K      0   419M
tank        1.09G  1.81T      0  6.15K      0   417M
tank        4.70G  1.81T      0  6.00K      0   402M
tank        4.70G  1.81T      0  6.49K      0   444M
tank        4.70G  1.81T      0  6.43K      0   433M
tank        4.70G  1.81T      0  6.41K      0   441M
tank        4.70G  1.81T      0  6.21K      0   426M
tank        8.53G  1.80T      0  5.72K      0   378M
tank        8.53G  1.80T      0  6.16K      0   408M
tank        8.53G  1.80T      0  6.25K      0   422M
tank        8.53G  1.80T      0  6.51K      0   444M
tank        8.53G  1.80T      0  5.95K      0   394M
tank        12.4G  1.80T      0  6.20K      0   420M
tank        12.4G  1.80T      0  6.01K      0   413M
tank        12.4G  1.80T      0  6.41K      0   435M
tank        12.4G  1.80T      0  6.54K      0   449M
tank        16.2G  1.80T      0  5.63K      0   370M
tank        16.2G  1.80T      0  6.10K      0   412M
tank        16.2G  1.80T      0  6.17K      0   420M
tank        16.2G  1.80T      0  6.38K      0   436M
tank        16.2G  1.80T      0  6.54K      0   453M
tank        20.1G  1.79T      0  5.66K      0   373M
tank        20.1G  1.79T      0  6.16K      0   411M
```

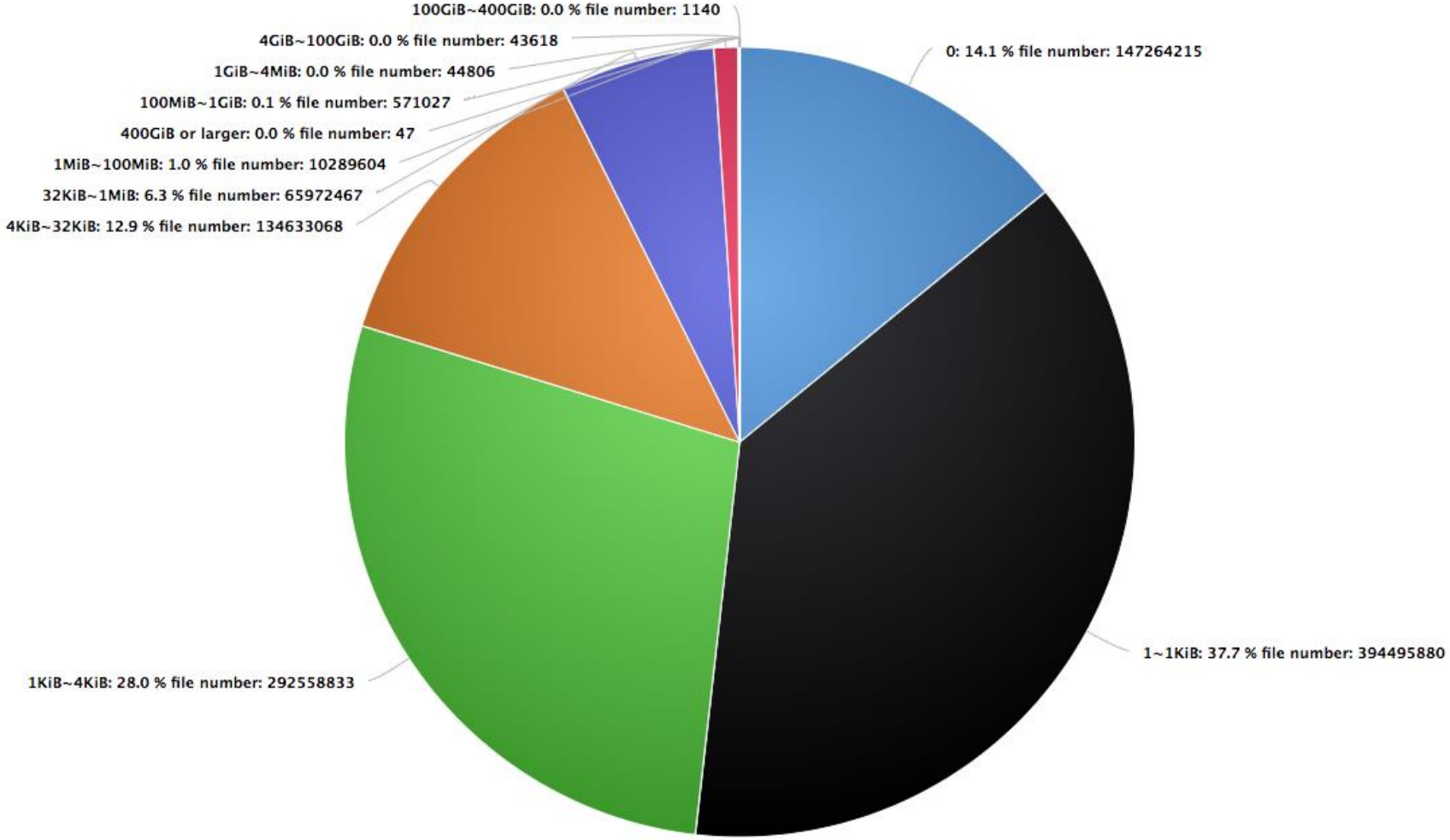
```
$ echo 0 > zio_dva_throttle_enabled
$ dd if=/dev/zero of=/tank/test1 bs=1M &
$ zpool iostat 2
              capacity     operations     bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
tank         101G  1.71T      1  4.38K  35.1K   303M
tank         104G  1.71T      0  15.0K  4.00K  1.70G
tank         108G  1.71T      0  14.8K      0  1.68G
tank         111G  1.70T      0  13.6K  4.00K  1.54G
tank         115G  1.70T      1  13.7K  7.99K  1.55G
tank         118G  1.70T      0  13.9K  4.00K  1.58G
tank         121G  1.69T      0  12.5K  4.00K  1.42G
tank         125G  1.69T      0  13.6K  4.00K  1.53G
tank         125G  1.69T      0  9.08K  4.00K  1.04G
tank         128G  1.69T      0  15.2K      0  1.73G
tank         132G  1.68T      0  14.2K  4.00K  1.61G
tank         135G  1.68T      0  12.8K  4.00K  1.45G
tank         139G  1.68T      2  13.7K  12.0K  1.56G
tank         142G  1.67T      0  13.8K  4.00K  1.57G
tank         146G  1.67T      0  13.2K  4.00K  1.50G
tank         149G  1.67T      0    13K  4.00K  1.47G
tank         153G  1.66T      2  13.3K  9.98K  1.50G
tank         153G  1.66T      3  13.7K  16.0K  1.56G
tank         156G  1.66T      0  15.4K      0  1.76G
tank         160G  1.66T      1  14.4K  6.00K  1.64G
tank         163G  1.65T      0  12.1K  4.00K  1.34G
tank         167G  1.65T      0  12.9K  4.00K  1.46G
tank         170G  1.65T      0  13.6K  4.00K  1.53G
```

zio_dva_throttle_enabled controls throttling of block allocations in the ZFS I/O (ZIO) pipeline.

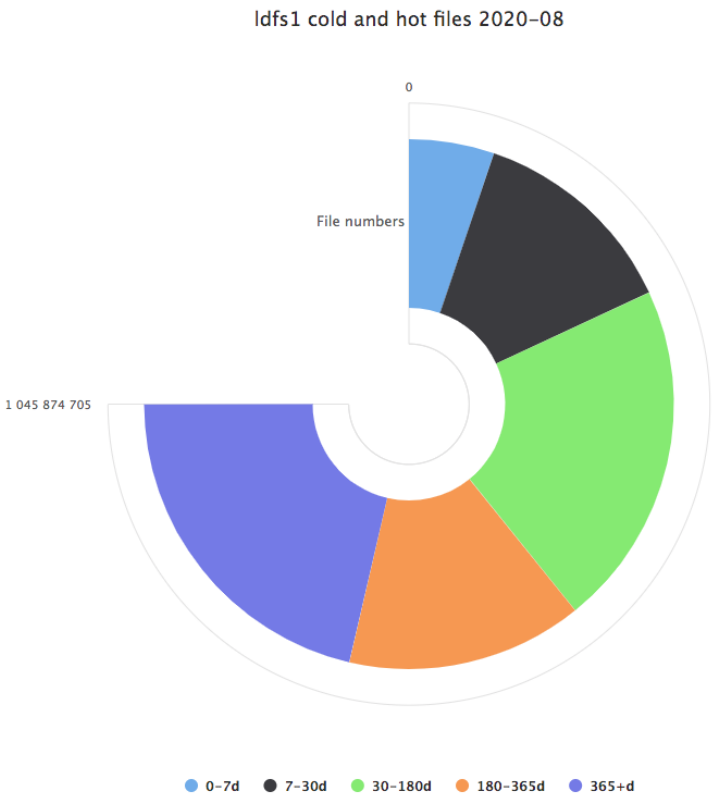
Work with Bioinformatics software

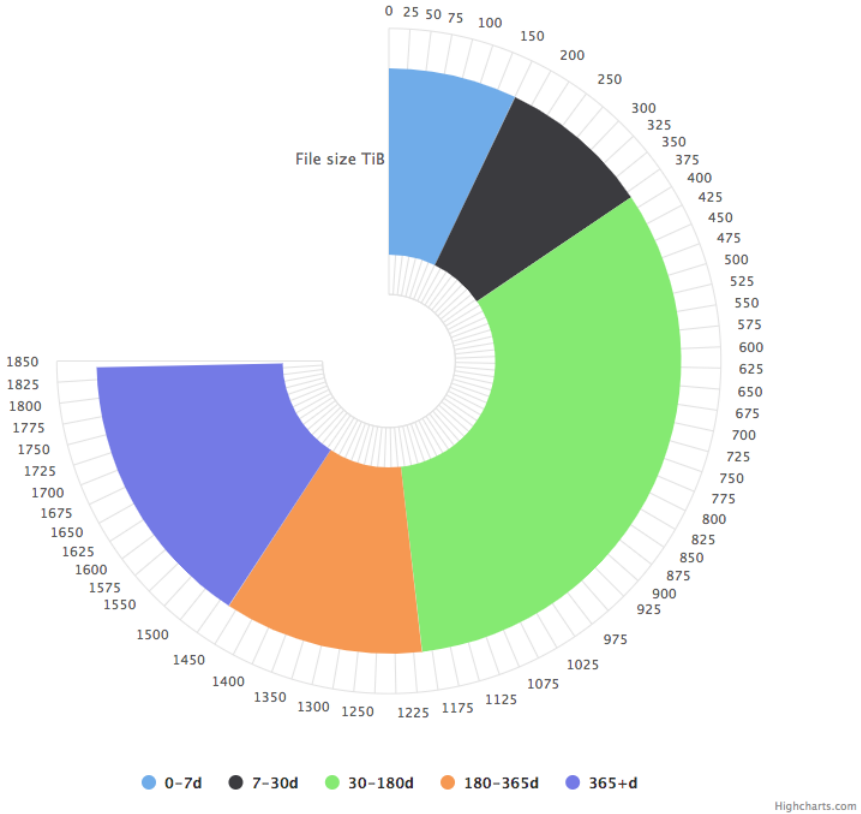
- Too many files (large than a billion in single namespace)
  - Tons of metadata stress
    - 10 millions files and dirs in single dir
      - No any sub-directory
      - Too many empty files and dirs
- The base unit is a “line”
  - That way too many random small IO
  - Eg: sort each line for a huge file
    - 4K/8K sequence/random(sort) I/O



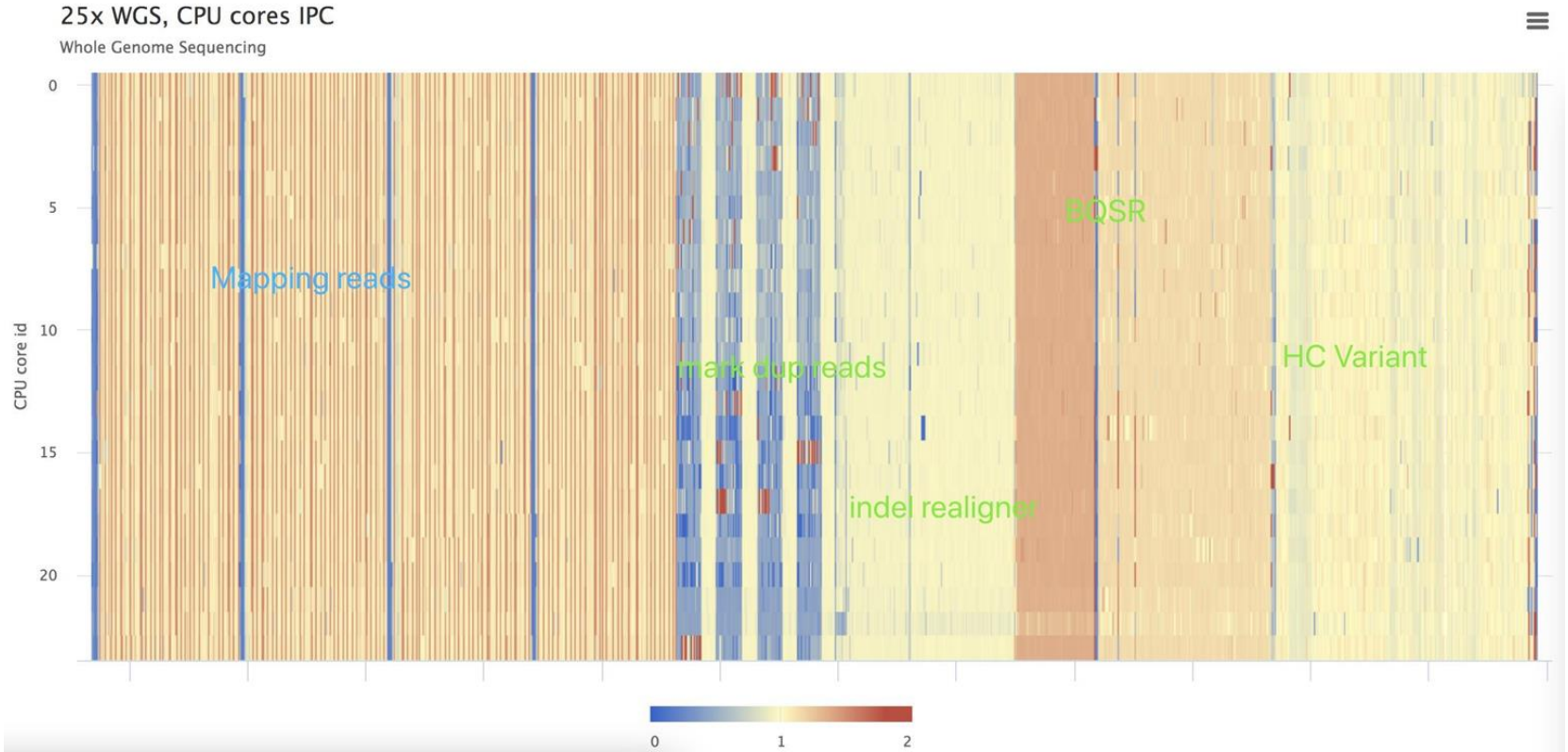

100GiB~400GiB: 0.0 % file number: 1140
4GiB~100GiB: 0.0 % file number: 43618
1GiB~4MiB: 0.0 % file number: 44806
100MiB~1GiB: 0.1 % file number: 571027
400GiB or larger: 0.0 % file number: 47
1MiB~100MiB: 1.0 % file number: 10289604
32KiB~1MiB: 6.3 % file number: 65972467
4KiB~32KiB: 12.9 % file number: 134633068
0: 14.1 % file number: 147264215
1~1KiB: 37.7 % file number: 394495880
1KiB~4KiB: 28.0 % file number: 292558833

Hot and cold files

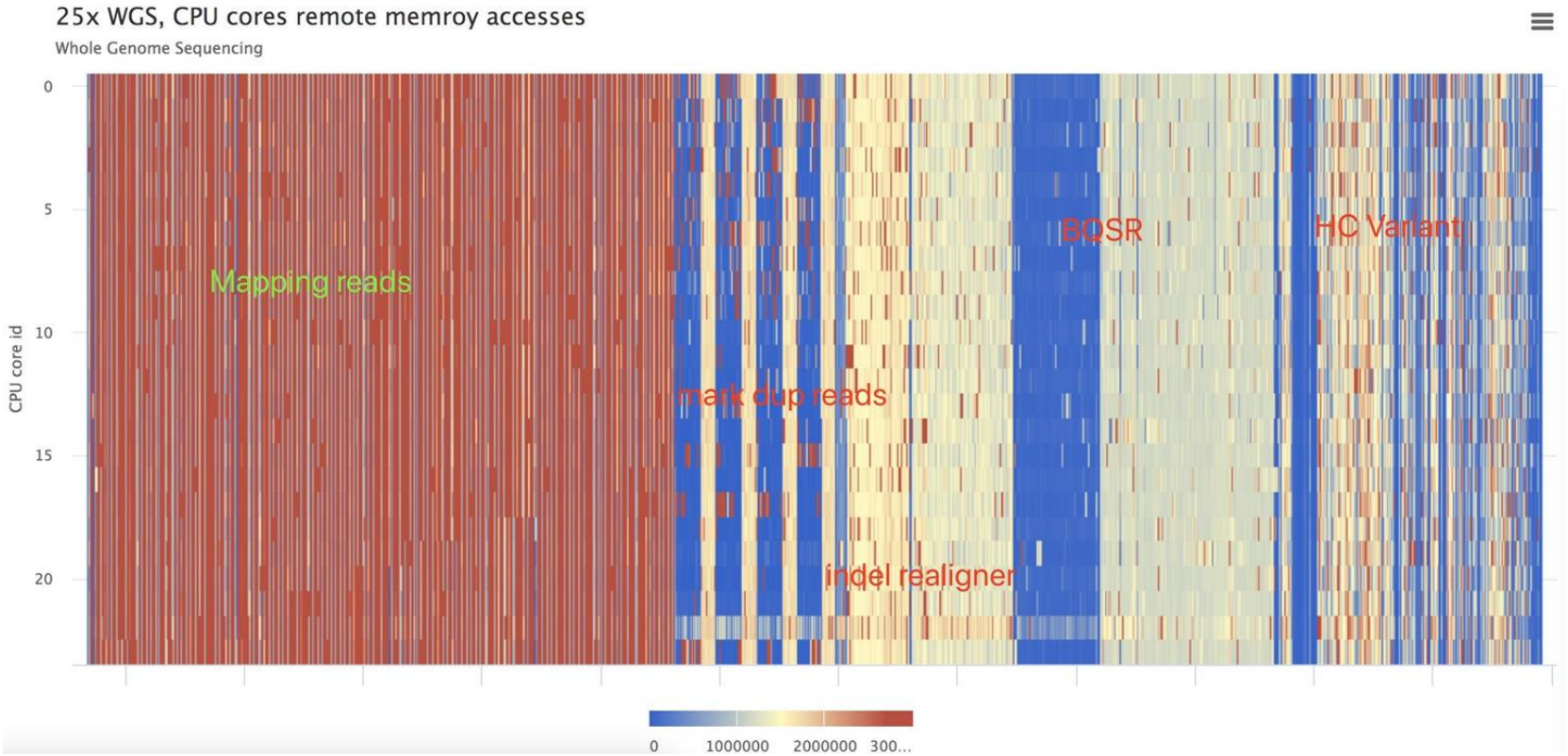


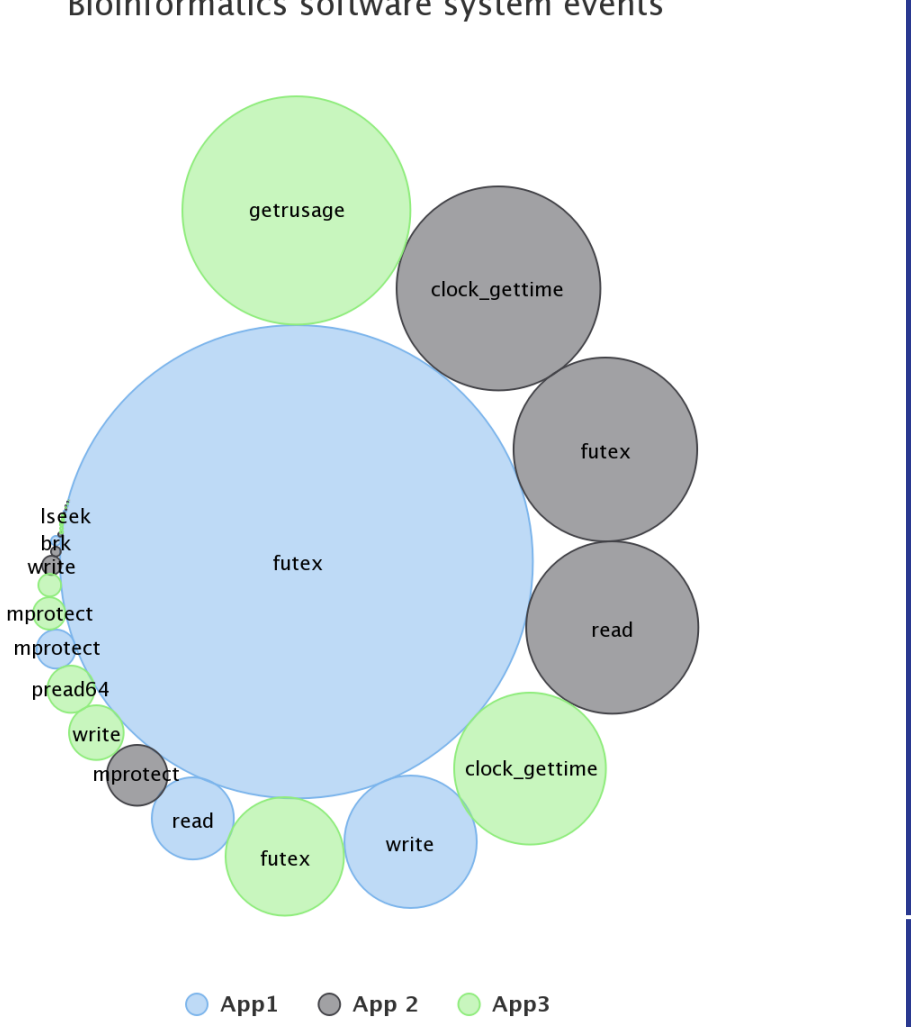


A BIO job CPU IPC



CPU access remote memory





- Trace system events
  - Need more optimize
    - Futex
    - Getrusage
    - Clock_gettime

PLY script for eBPF

```
#!/usr/local/sbin/ply
kprobe:sys_unlinkat / !strcmp(execname, "rm") /
{
   t[pid] =  walltime;
   printf("pid: %d %s\n", pid, str(arg1) );
}

kretprobe:sys_unlinkat / !strcmp(execname, "rm") /
{
   printf("sys_unlinkat take time: %lld ns\n", walltime - t[pid] );
}
```

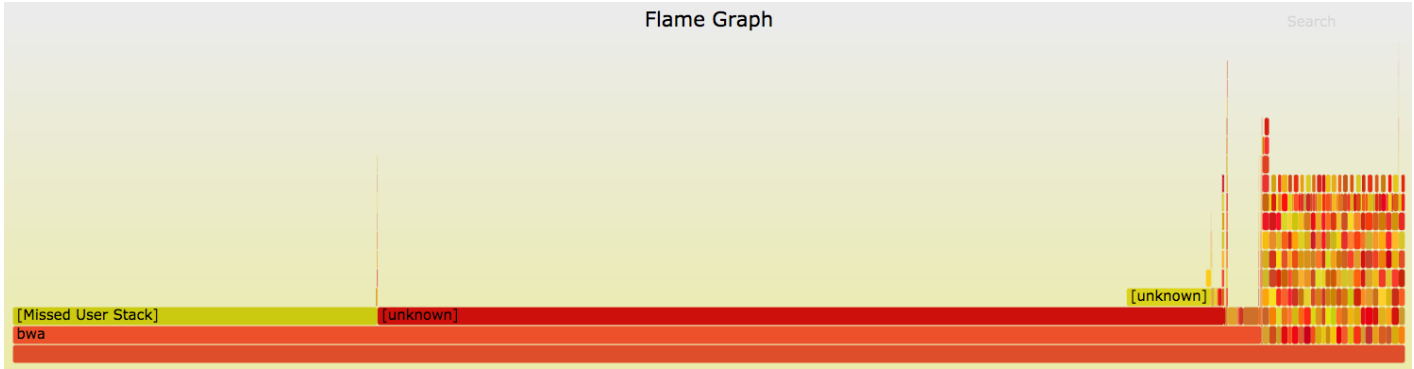
```
pid: 27150 test_6d61c81d1
sys_unlinkat take time: 1483215 ns
pid: 27135 test_5d4a93ff2
sys_unlinkat take time: 1484306 ns
pid: 27150 test_6d61c81d1
sys_unlinkat take time: 1506183 ns
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
27135 root      20   0  115236   7780    580 D   6.2  0.0  14:36.30 rm
27150 root      20   0  115164   7708    580 D   6.2  0.0  25:40.93 rm
```

The eBPF tool is available as a Technology Preview in RHEL 7.6

Ply is easy to use and install, lightweight

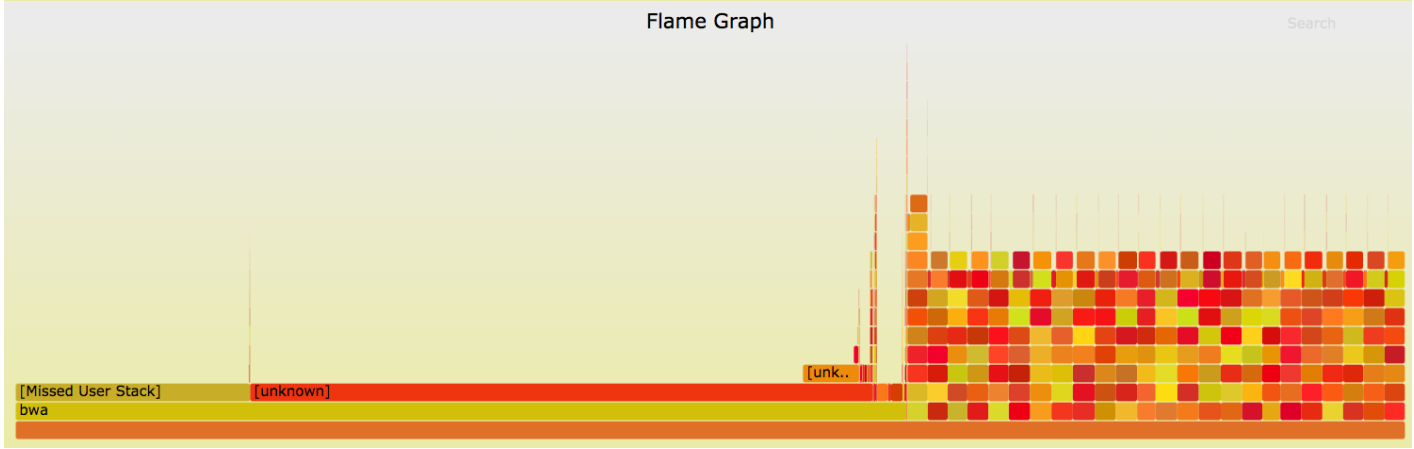
Profile-bpfcc output flamegraph



/usr/share/bcc/tools/profile -f 120 > out.folded

cat out.folded | ./flamegraph.pl > test.svg

Mapping reads
← pin local numa resource (300s)
12 threads in 12 threads in a node



Mapping reads
← force process fight(300s)
24 threads in 12 threads in a node
(Simulated user actions, Our users like more threads fight)

Monitor

Enable ZFS debug message log
echo 1 > /sys/module/zfs/parameters/zfs_dbgmsg_enable
/proc/spl/kstat/zfs/dbgmsg

/proc/spl/kstat/zfs/$pool_name
arcstat,arc_summary,zpool iostat -lv,-w,-r, zdb(powerful)……

When 5s(default config) pass without a successful MMP write in any device, the suspended will coming. You have to power reset
99.9% zpool suspended because single HDD write failed

zpool import protection offered by MMP without the concern that it might suspend your pool
Set zfs_multihost_fail_intervals = 0 (Test carefully with 0.7.13)

SAS PHY signal:
Running disparity,phy reset,loss of dword sync,Invalid dword

```
                capacity     operations     bandwidth    total_wait     disk_wait    syncq_wait    asyncq_wait  scrub   trim
pool         alloc   free   read  write   read  write   read  write   read  write   read  write   read  write   wait    wait
----------  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----
tank         51.7T  37.4T    303     82  9.08M  9.70M   41ms  164ms    4ms    7ms   28ms  106us   44ms  173ms    2ms      -
  raidz2     51.7T  37.4T    303     82  9.08M  9.70M   41ms  164ms    4ms    7ms   28ms  106us   44ms  173ms    2ms      -
    sda          -      -     30      8   930K   993K   43ms  164ms    4ms    7ms   29ms  106us   46ms  174ms  898us      -
    sdb          -      -     30      8   929K   993K   35ms  161ms    4ms    7ms   20ms   93us   41ms  171ms  726us      -
    sdc          -      -     30      8   928K   993K   35ms  168ms    4ms    8ms   19ms   97us   41ms  177ms    2ms      -
    sdd          -      -     30      8   927K   993K   33ms  159ms    4ms    7ms   17ms   96us   39ms  169ms    2ms      -
    sde          -      -     30      8   931K   993K   47ms  163ms    4ms    7ms   38ms  108us   48ms  173ms    2ms      -
    sdf          -      -     30      8   931K   993K   50ms  165ms    4ms    7ms   42ms  106us   49ms  174ms    2ms      -
    sdg          -      -     30      8   928K   993K   32ms  162ms    4ms    7ms   16ms  116us   39ms  171ms    1ms      -
    sdh          -      -     30      8   930K   993K   40ms  165ms    4ms    7ms   26ms  111us   44ms  175ms    4ms      -
    sdi          -      -     30      8   930K   993K   41ms  163ms    4ms    7ms   28ms  108us   45ms  173ms    2ms      -
    sdl          -      -     30      8   932K   993K   54ms  167ms    4ms    8ms   48ms  122us   50ms  176ms    4ms      -
----------  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----  -----
```

```
NAME        STATE     READ WRITE CKSUM  temp
tank        ONLINE       0     0     0
  raidz2-0  ONLINE       0     0     0
    sda     ONLINE       0     0     0    29
    sdb     ONLINE       0     0     0    29
    sdc     ONLINE       0     0     0    29
    sdd     ONLINE       0     0     0    29
    sde     ONLINE       0     0     0    29
    sdf     ONLINE       0     0     0    28
    sdg     ONLINE       0     0     0    29
    sdh     ONLINE       0     0     0    29
    sdi     ONLINE       0     0     0    29
    sdl     ONLINE       0     0     0    29
```

```
NAME        STATE     READ WRITE CKSUM
    sdjh    ONLINE       0   135     0
```

```
+-host14
    \-expander-14:12  loss_of_dword_sync(4,4,4,4,4,4,4,4)  phy_reset(0,0,0,0,0,0,0,0)
        |   \-enclosure  [14:0:409:0]  /dev/sg285  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot00:[14:0:401:0]  /dev/sdjh  /dev/sg277  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot01:[14:0:402:0]  /dev/sdji  /dev/sg278  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot02:[14:0:403:0]  /dev/sdjj  /dev/sg279  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot03:[14:0:404:0]  /dev/sdjk  /dev/sg280  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot04:[14:0:405:0]  /dev/sdjl  /dev/sg281  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot05:[14:0:406:0]  /dev/sdjm  /dev/sg282  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot06:[14:0:407:0]  /dev/sdjn  /dev/sg283  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot07:[14:0:408:0]  /dev/sdjo  /dev/sg284  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot08:[14:0:375:0]  /dev/sdih  /dev/sg251  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot09:[14:0:376:0]  /dev/sdii  /dev/sg252  loss_of_dword_sync(0)  phy_reset(0)
        |       +-sg285 Slot10:[14:0:377:0]  /dev/sdij  /dev/sg253  loss_of_dword_sync(0)  phy_reset(0)
        |       \-sg285 Slot11:[14:0:378:0]  /dev/sdik  /dev/sg254  loss_of_dword_sync(0)  phy_reset(0)
```

Optimization and monitor

/proc/spl/kstat/zfs/dmu_txg
/proc/spl/kstat/zfs/tank/txgs
/proc/spl/kstat/zfs/arcstats

Lustre not support ZIL，In the high loading，ZFS OPEN frequently
Increase value for /sys/module/zfs/parameters/zfs_dirty_data_sync, dmu_dirty_delay and dmu_tx_dirty_over_max count not rise, the sync interval be better
it depends the write loading

Too many small write serially in bioinformatics software
Enable zfs_prefetch_disable helps us
echo 0 > /sys/module/zfs/parameters/zfs_prefetch_disable

```
cat /proc/spl/kstat/zfs/dmu_tx
12 1 0x01 11 2992 23459648759 6610514898846074
name                            type data
dmu_tx_assigned                 4    562496257
dmu_tx_delay                    4    0
dmu_tx_error                    4    0
dmu_tx_suspended                4    0
dmu_tx_group                    4    374305
dmu_tx_memory_reserve           4    0
dmu_tx_memory_reclaim           4    0
dmu_tx_dirty_throttle           4    15665
dmu_tx_dirty_delay              4    19537866
dmu_tx_dirty_over_max           4    4394762
dmu_tx_quota                    4    0
```

Why the zfs default parameters limit zpool performance ?

```
1258    798040694884157  C       100876288       1144877056      60624896        81640   2454    1611384656      45344513        31210   837186819
1259    798042306268813  C       933888          908152832       44560384        64639   1974    45382470        3715    837192289       634480093
1260    798042351651283  C       79511552        1108844544      55009280        78995   2307    1471685298      462125  18607   767231466
1261    798043823336581  C       671744          784834560       38649856        55974   1751    484174  3873    767240116       575089847
1262    798043823820755  C       71909376        1167433728      70070272        83228   2854    1342342696      784361  29173   865772575
1263    798045166163451  C       720896          891023360       43347968        63477   1973    818363  3806    865779417       624814411
1264    798045166981814  C       87638016        1154531328      64962560        82199   2683    1490606691      31614821        37554   810320650
1265    798046657588505  C       933888          1012711424      54947840        72122   2458    31658235        3602    810327480       715761187
1266    798046689246740  C       383598592       898265088       48177152        63993   2070    5810209206      3220087 31366   677728916
1267    798052499455946  C       326451200       970227712       60854272        69148   2531    5003992534      2166892 40226   721786925
1268    798057503448480  C       336150528       980021248       58585088        69863   2476    5002991486      6307086 27957   732857595
1269    798062506439966  C       347684864       1162309632      66641920        82793   2804    5005994200      14603912        32488   853161354
1270    798067512434166  C       330776576       1017225216      61075456        72463   2598    5013996549      34260276        32834   754878937
1271    798072526430715  C       323960832       1083658240      67092480        77178   2828    5033993619      29897335        29043   792485200
1272    798077560424334  C       325533696       986927104       64942080        70425   2789    5028994745      1441237 30916   850262275
1273    798082589419079  C       331300864       1016184832      64110592        72366   2749    5000998649      13924634        53057   801395772
```

Disabled in most of firmware

- WCE (Write Cache Enable)
  - bit 0 SCSI WRITE commands may not return status and completion message bytes until all data has been written to the media.
  - 1 SCSI WRITE commands may return status and completion message bytes as soon as all data has been received from the host.
  - 4K randwrite 3x IOPS
- EN_BMS (Enable Background Medium Scan)
  - Bit 0 An enable background medium scan (EN_BMS) bit set to zero specifies that background medium scan is disabled.
  - 1 An EN_BMS bit set to one specifies that background medium scan operations are enabled. If the EN_PS bit is also set to one then a background medium scan operation shall not start until after the pre-scan operation is halted or completed.
  - Reduce device life, start/stop continual
    - Help check error and generate new error
    - It always loop running, and not exit for
  - Replace it by scrub and smart test

Table 227 — Caching Parameters page (08h)

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	PAGE CODE (08h)					
1	PAGE LENGTH (12h)							
2	IC	ABPF	CAP	DISC	SIZE	WCE	MF	RCD

Table 226 — Background Control mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(1b)	PAGE CODE (1Ch)					
1	SUBPAGE CODE (01h)							
2	(MSB)	PAGE LENGTH						
3								(LSB)
4	Reserved							EN_BMS
5	Reserved							EN_PS

```
# smartctl -l background /dev/sdb
smartctl 6.2 2013-07-26 r3841 [x86_64-linux-3.10.0-957.el7_lustre.x86_64] (local build)
Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
Background scan results log
  Status: scan is active
    Accumulated power on time, hours:minutes 29315:49 [1758949 minutes]
    Number of background scans performed: 91,  scan progress: 92.02%
    Number of background medium scans performed: 91
```

Clean tiny files

- ➔ Only scan directory
  - ◆ Record size of directory to make sure the huge directory
  - ◆ All scan stress on Lustre MDT(SSD), offload find IO in each OST
  - ◆ Find speed improve 30x
- ➔ File system changelog
  - ◆ Lustre changelog unstable

```
  File: [redacted]
  Size: 1171489280      Blocks: 2288065    IO Block: 131072 directory
Device: be1c918eh/3189543310d   Inode: 1441217[redacted]233029705  Links: 2
Access: [redacted]              Uid: [redacted]    Gid: [redacted]
Access: 2020-09-22 17:40:33.000000000 +0800
Modify: 2020-09-21 10:43:26.000000000 +0800
Change: 2020-09-21 10:43:26.000000000 +0800
```

ZFS highlight

- Perfect throughput
- Easy to manage with different manufacturers
- The cheapest price
- More readily automated/monitor
- ZFS is a lighthouse for a lot of open-source and private project, A real Enterprise features and open-source file system

OpenZFS, Ext4 in the production

	OpenZFS	Ext4/ldiskfs
Online scrub	Y	N
Checksum block data	Y	N, only metadata, deps the hardware
Crash consistency	Y	N, Full fsck time ?
Online async replication	Y	N
Compression	Y (lz4 1.02x ~ 1.20x in different environment)	N
Performance	Lustre does not support zfs zil or another write cache	balance
Software manageability	Easy (resolved all maintain issues)	Easy
Hardware	IO Expander management module no detail log, no full support for T10 ses	Mature enough in SAN vendor
Crash ratio	zfs 0.7.x (MMP, high) > zfs 0.6.5 (low)	Low
Cost-effective	The cheapest price, be good at throughput	costly, limit the performance

All issues just in the ZOL open-source version, not sure in the others fork version

ZOL_issues #2831#2449 #4877 #7068 #7709 #7731 #7834 #8495 #10018 #10838 #10873

With special thanks to a zfs developer’s powerful help

2020-09