ZFS on illumos

Prakash Surya
Where ZFS originated

- 2001: Started at Sun
- 2005: Released through OpenSolaris
- 2010: illumos spawned, fork of OpenSolaris
- 2013: OpenZFS created
- ZFS’s “home” is in illumos:
  - Due to its history, but also its OS integration: grub, mdb, fma, etc
- But, OpenZFS is growing beyond illumos
Development model on illumos

- Committer access is granted to “advocates”
- Advocates rely on “reviewers” to verify changes for correctness, good design, etc.
- No explicit releases
  - All changes must be “release quality”
- Development tools/processes are difficult
  - e.g. patch/compile/deploy/test is cumbersome
How to facilitate collaboration?

- We encourage “upstreaming” changes
  - Difficult with current development model
- How can we make collaboration easier?
  - We’re open to changes in development model
    - Peer code reviews are good
    - High overhead to build and test is bad
- Would an OpenZFS repository help?
  - If so, what are the requirements?
  - How can we get there?
Perspective coming from ZOL

- Large overhead for ZFS on illumos changes
  - ZFS on illumos is tightly integrated with illumos
    - illumos is the kernel, libraries, and more
  - Overhead for “lone” developer is prohibitive
  - ZFS on Linux is isolated, little dependencies
- Full illumos build: ~2 hours
  - Building ZFS only: ~6 minutes
- ZFS on Linux build time: ~3 minutes
ZOL to illumos continued

- Kernel tools are generally much better
  - mdb is awesome! crash probably could be.
    - pipelines and walkers
      - “SQL for crash dumps”
    - dcmds allow extensibility
      - ZFS specific extensions
  - ::walkers, ::findleaks, ::stacks -m zfs, ::whatis, ::spa, ::dbuffs, ::blkptr, ::zio_state
- No gdb; no line number resolution
- kmdb and dtrace are also very helpful
ZOL to illumos continued

- Smaller community of ZFS users on illumos
  - People involved are more informed
  - Fewer number of people testing
- ZFS test suite available on illumos
  - But, no xfstests or filebench
# mdb example - ::spa -v

```bash
> ::spa -v ! head -n 15
ADDR             STATE NAME
ffffff096151a000  ACTIVE rpool
```

<table>
<thead>
<tr>
<th>ADDR</th>
<th>STATE</th>
<th>AUX</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>fffffff095050c780</td>
<td>HEALTHY</td>
<td>-</td>
<td>root</td>
</tr>
<tr>
<td>fffffff09505106c0</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c2t0d0s0</td>
</tr>
<tr>
<td>fffffff09630ac000</td>
<td>ACTIVE</td>
<td></td>
<td>tank</td>
</tr>
<tr>
<td>fffffff096be74540</td>
<td>HEALTHY</td>
<td>-</td>
<td>root</td>
</tr>
<tr>
<td>fffffff09616f34c0</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c3t0d0s0</td>
</tr>
<tr>
<td>fffffff09629c9780</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c3t1d0s0</td>
</tr>
<tr>
<td>fffffff096be6f900</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c3t2d0s0</td>
</tr>
<tr>
<td>fffffff096be6f280</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c3t3d0s0</td>
</tr>
<tr>
<td>fffffff096be6ec00</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c3t4d0s0</td>
</tr>
<tr>
<td>fffffff096be6e580</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c3t5d0s0</td>
</tr>
</tbody>
</table>
mdb example - ::spa -Mh

> ::spa -Mh ! head -n 15

<table>
<thead>
<tr>
<th>ADDR</th>
<th>STATE</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffffffff096151a000</td>
<td>ACTIVE</td>
<td>rpool</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADDR</th>
<th>STATE</th>
<th>AUX</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffffffff095050c780</td>
<td>HEALTHY</td>
<td>-</td>
<td>root</td>
</tr>
<tr>
<td>ffffffff09505106c0</td>
<td>HEALTHY</td>
<td>-</td>
<td>/dev/dsk/c2t0d0s0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADDR</th>
<th>FRAGMENTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffffffff095986b740</td>
<td>32%</td>
</tr>
</tbody>
</table>

9: 113 **********
10: 131 ***********
11: 391 ***********************************
12: 456 ****************************************
13: 250 ********************
14: 227 *********************
15: 386 ###########################
mdb example - ::dbufs

> ::dbufs ! wc -l
182819

> ::dbufs | ::print dmu_buf_impl_t ! head -n 15
{

db = {
    db_object = 0x76
    db_offset = 0x1a4a0000
    db_size = 0x20000
    db_data = 0xffffff03b2dcd000
}
db_objset = 0xffffff0991377c00
db_dnode_handle = 0xffffff09e0266d58
db_parent = 0xffffff09e4b22808
db_hash_next = 0
db_blkid = 0xd25
db_blkptr = 0xffffff09e21a5280
db_level = 0
db_mtx = {

mdb example - ::dbuf

```bash
> ::dbufs | ::dbuf ! head -n 15

<table>
<thead>
<tr>
<th>addr</th>
<th>object</th>
<th>lvl</th>
<th>blkid</th>
<th>holds</th>
<th>os</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffffffff0af2001010</td>
<td>76 0</td>
<td>d25</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c26001018</td>
<td>84 0</td>
<td>68c</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c260010f8</td>
<td>77 0</td>
<td>1e9</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0af20011d0</td>
<td>71 0</td>
<td>dc5</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c260011d8</td>
<td>65 0</td>
<td>b55</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0af20012b0</td>
<td>7e 0</td>
<td>fb8</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c260012b8</td>
<td>80 0</td>
<td>a8a</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c26001398</td>
<td>b7 0</td>
<td>a2b</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0af2001470</td>
<td>6e 0</td>
<td>91e</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c26001478</td>
<td>86 0</td>
<td>834</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0af2001550</td>
<td>85 0</td>
<td>e05</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c26001558</td>
<td>87 0</td>
<td>851</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0af2001630</td>
<td>6a 0</td>
<td>353</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
<tr>
<td>ffffffff0c26001638</td>
<td>74 0</td>
<td>49d</td>
<td>0</td>
<td>tank/fish</td>
<td></td>
</tr>
</tbody>
</table>
```
mdb example - ::whatis

> `ffffffff09e4b22808::whatis` ! head -n 15

`ffffffff09e4b22808` is **allocated from `dmu_buf_impl_t`**:

<table>
<thead>
<tr>
<th>ADDR</th>
<th>BUFADDR</th>
<th>TIMESTAMP</th>
<th>THREAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffffffffff09e50de9c0</td>
<td>ffffffffff09e4b22808</td>
<td>3ed28b4787</td>
<td>ffffffffff09beccc840</td>
</tr>
<tr>
<td>ffffffffff095b1c0448</td>
<td>ffffffffff090f2b6900</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- kmem_cache_alloc_debug+0x2e0
- kmem_cache_alloc+0x2d0
- dbuf_create+0x5a
- dbuf_hold_impl+0x177
- dbuf_findbp+0x17b
- dbuf_hold_impl+0xf9
- dbuf_hold_level+0x31
- dbuf_hold+0x21
- dmu_buf_hold_array_by_dnode+0x109
- dmu_read_uio_dnode+0x5a
THANK YOU
ANY QUESTIONS?