CEPHALOCON APAC 2018

THE FUTURE OF STORAGE

22-23 March 2018 | BEIJING

Linux block cache practice on Ceph BlueStore

Junqin Zhang

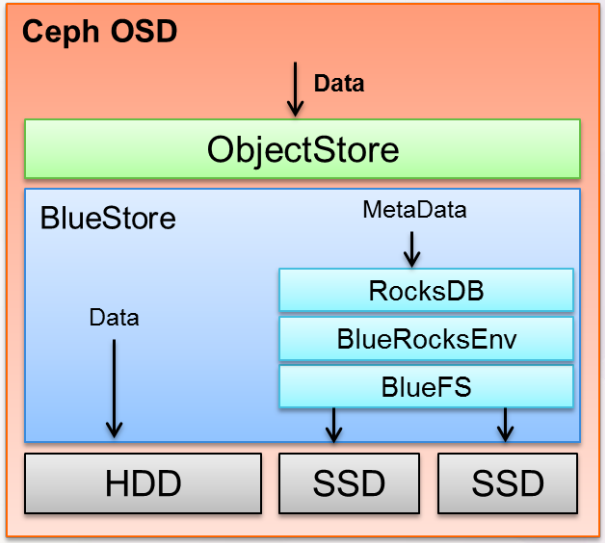
Lenovo Cloud Technology Center

Contents

- Ceph BlueStore
- Linux Block Cache
- BlueStore on Linux block cache
- Found problems
- Future direction suggestion

Ceph BlueStore

- BlueStore is a new storage backend for Ceph. It is the default storage backend for Ceph OSDs in Luminous v12.x.
- BlueStore allows for storing objects directly on the Ceph block devices without any file system interface.
- BlueStore can manage up to three devices: main device, db device, WAL device.
- Ceph BlueStore has a overall better and more stable performance than Ceph FileStore.

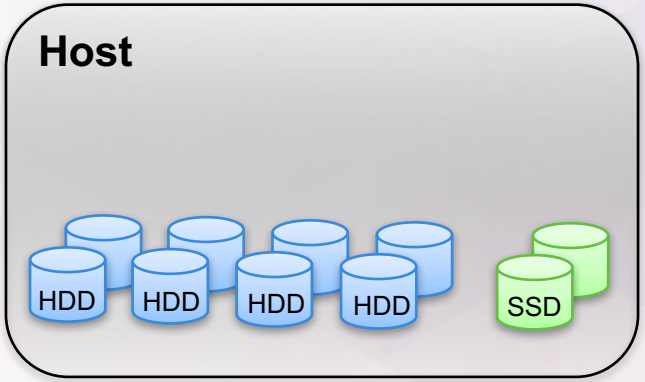


Fast disk utilization in BlueStore

- Even though BlueStore is generally able to make much better use of the fast device and use more space than FileStore, it has many fast device space left in some hardware environment.

- Take our hardware environment for example, in each host:
  - SATA HDD 6T *8
  - SATA SSD 800G *2
- If each OSD allocates 100G for DB and WAL device, there are still 800G left. And if provision very big DB device at beginning, then there would be many fast device space not used in not high cluster usage.

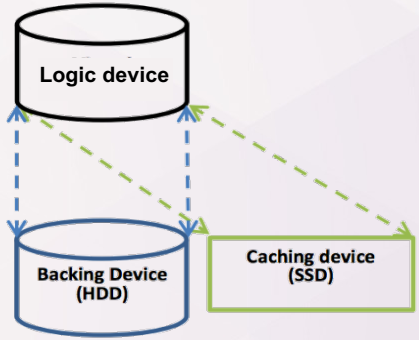
- To fully utilize fast device, we use Linux block cache to improve HDD performance.



Linux Block Cache

ceph

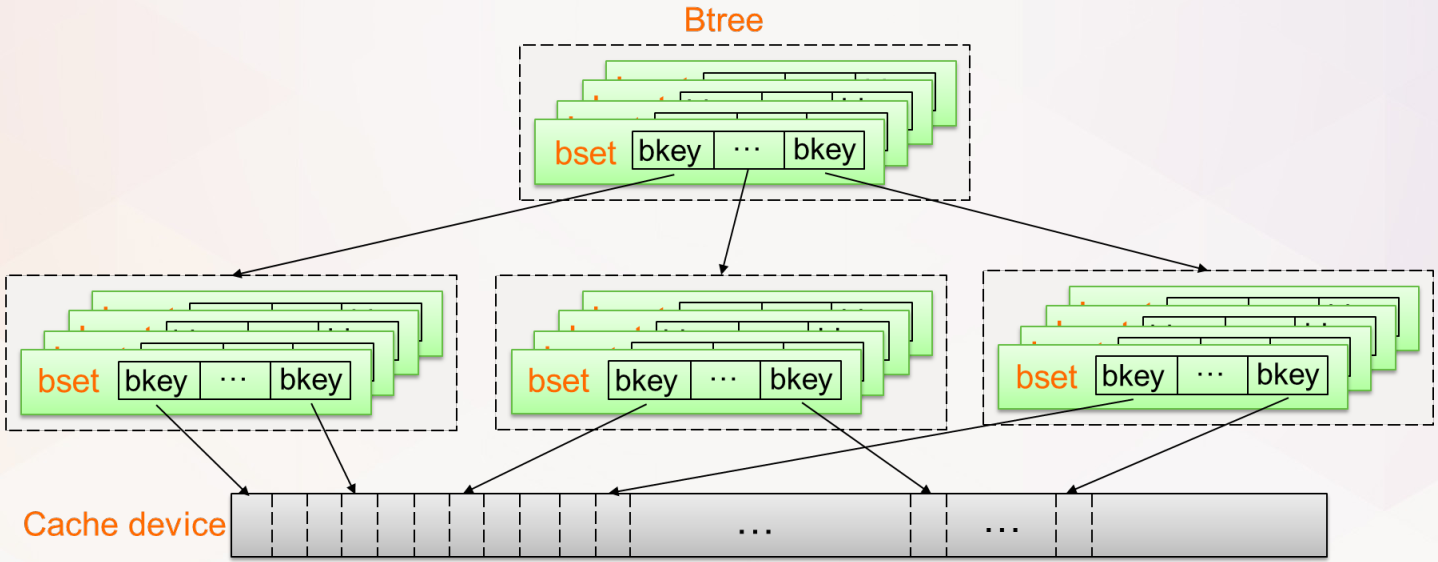
- Linux block cache solution allows one or more fast disk drives such as SSD to act as a cache for slower hard disk drives.
- A logical device is presented to the file-system (or applications) instead of the actual destination HDD where data was meant to be stored.
- There are several open source Linux block cache solutions.



LVM Cache	Bcache	DM-writeboost
• Merged to kernel 3.9 • Based on Linux kernel's device mapper • Use logical volumes to setup DM-Cache • Low performance	• Merged to kernel 3.10 • Based on block device layer • Designed around performance characteristics of SSDs • Many features/configuration options	• Based on Linux kernel's device mapper • Log-structured caching, control three layers(RAM buffer, caching device and backing device) • Friendly usage tools set

Bcache introduction

- Use B+ tree to manage one or more cache devices which are split as buckets.
- Use hash table to save Btree nodes to improve lookup performance. And use journal to improve Btree nodes updating performance.
- Random writes are turned into sequential writes to HDD by using SSD as buffer.



Bcache introduction

- Allows one or more fast disk drives to act as a cache for one or more slower hard disk drives.

- Support cache mode:
  - write through, write back, write around

- Support cache replacement policy
  - LRU, FIFO or Random

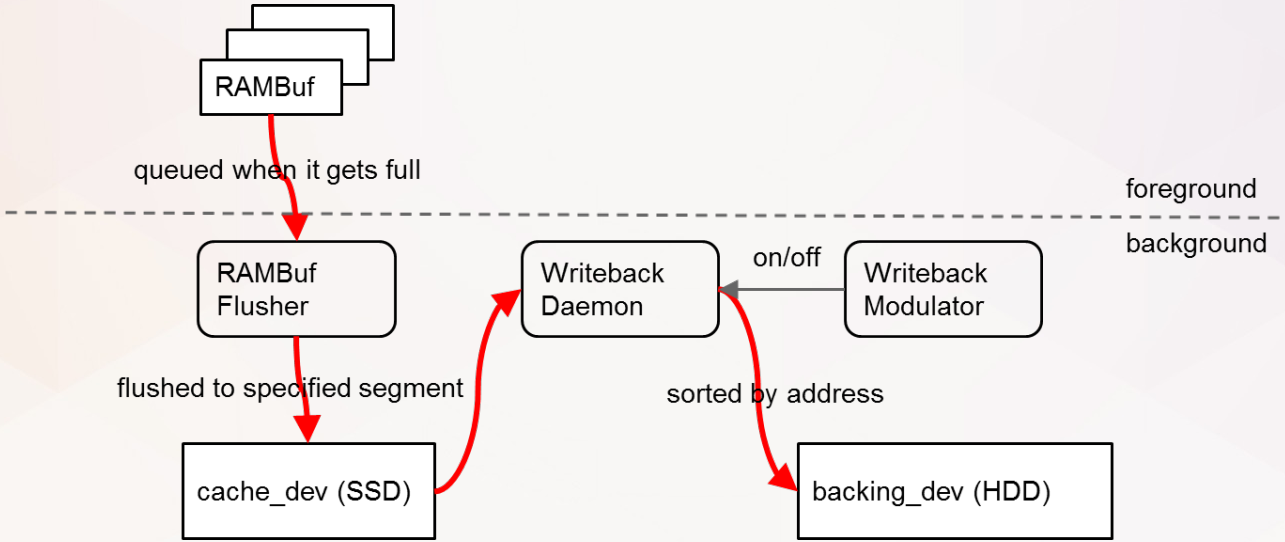
- Support discard/TRIM, default to off

```
NAME                MAJ:MIN RM   SIZE RO TYPE
sda                   8:0    0 745.2G  0 disk
└─sda1                8:1    0   200G  0 part
  ├─bcache0         252:0    0   5.5T  0 disk
  └─bcache1         252:1    0   5.5T  0 disk

sdd                   8:48   0   5.5T  0 disk
└─bcache0           252:0    0   5.5T  0 disk
sde                   8:64   0   5.5T  0 disk
└─bcache1           252:1    0   5.5T  0 disk
```

DM-writeboost introduction

- Control three different layers RAM buffer, caching device and backing device.
- Build logs from in-coming writes (data and metadata) and then writes the logs sequentially similar to log-structured file system.
- Use chained hash table to look up data in cache device.



DM-writeboost introduction

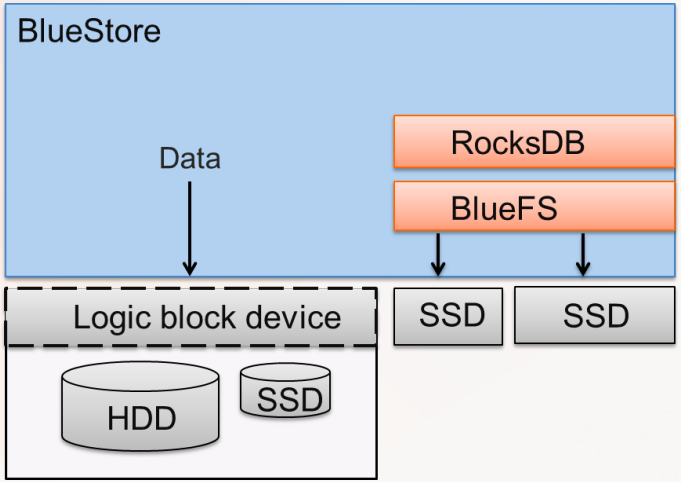
- Only allows one fast disk drive to act as a cache for one slower hard disk drive.
- Support cache mode:
  - write back, write around
- Support cache replacement policy:
  - FIFO
- Not support discard/TRIM

```
NAME          MAJ:MIN RM  SIZE RO TYPE
sda             8:0    0  745.2G 0 disk
├─sda1          8:1    0  100G   0 part
│ └─wbdev0      252:0  0  5.5T   0 dm
├─sda2          8:1    0  100G   0 part
│ └─wbdev1      252:1  0  5.5T   0 dm

sdd             8:48   0  5.5T   0 disk
└─ wbdev0       252:0   0  5.5T   0 dm
sde             8:64   0  5.5T   0 disk
└─ wbdev1       252:1   0  5.5T   0 dm
```

BlueStore on block cache

- BlueStore based on Linux block cache deployed as below:
  - DB and WAL are written to SSD directly.
  - A Logic block device is created by combined HDD and SSD, SSD is used as cache of HDD.
  - BlueStore OSD write data to logical device instead of HDD.




```
[osd.#]
host = host-name
osd data = /var/lib/ceph/osd/ceph-#
bluestore block wal path = /dev/ssd1
bluestore block db path = /dev/ssd2
bluestore block path = /dev/logic-device
```

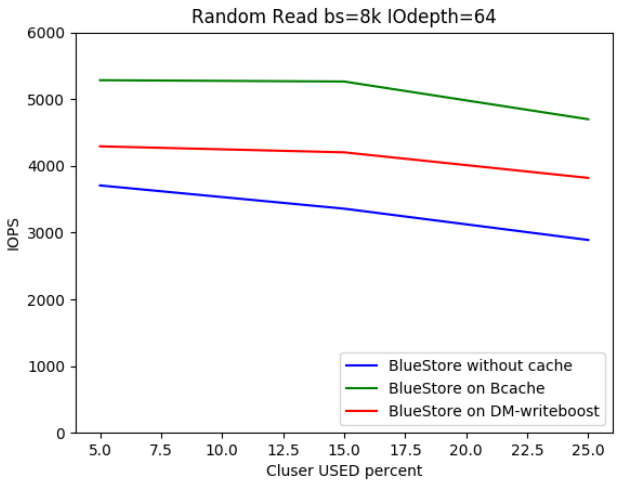
Performance test

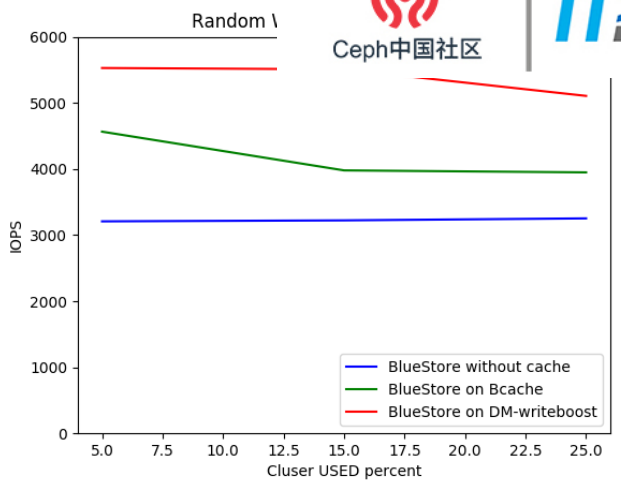
- Here we just test Ceph RBD performance.
- Test BlueStore OSD on Bcache and DM-writeboost and set to write back mode.
- Create RBDs and fill with data before test, then test with Fio.
- Test performance in different cluster data usage percent.

```
# ceph df
GLOBAL:
    SIZE     AVAIL     RAW USED     %RAW USED
    131T      130T        918G         0.68
POOLS:
    NAME     ID     USED     %USED     MAX AVAIL     OBJECTS
    rbd      1      19               0        124T           2
```

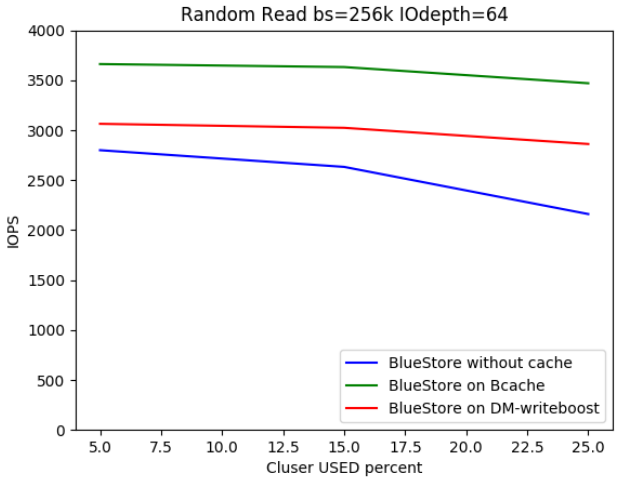
ceph

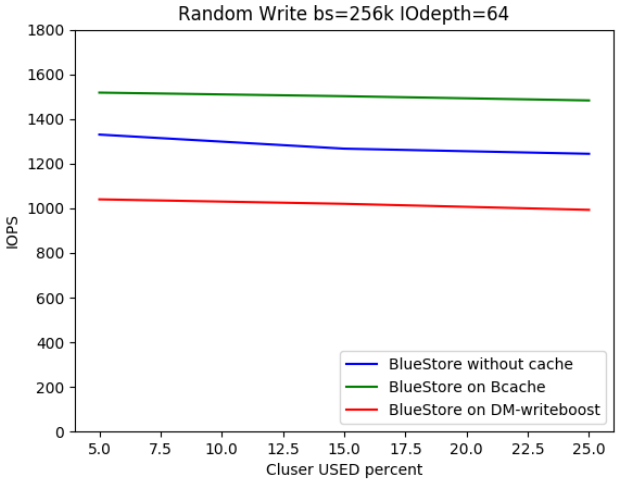
8K





256K





Performance test result

- Both BlueStore on Bcache and DM-writeboost have a better performance than Bluestore without cache.
- DM-writeboost is better at small IO write, but works bad on big IO write.
- Bcache is overall a better cache solution for BlueStore.

- Both BlueStore on Bcache and DM-writeboost have bigger IOPS variance than BlueStore without cache.
- DM-writeboost consumes more memory due to use memory as buffer.

Found problems

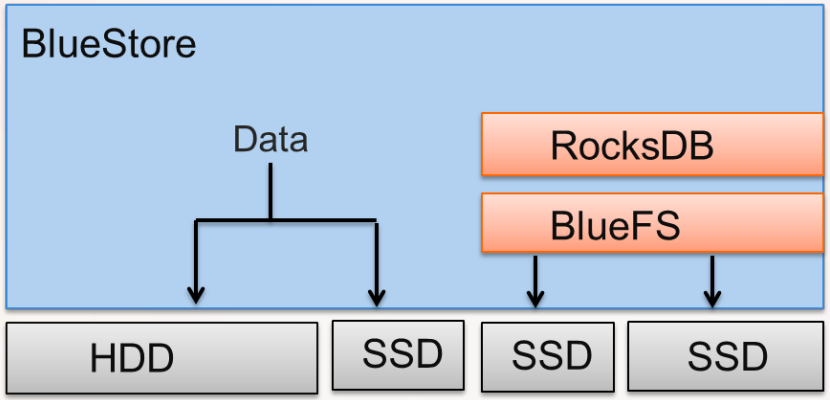
- ‘Slow Request’ in OSD when use same SSD disk for BlueStore and Cache
  - When test with high IO depth, both BlueStore on Bcache and DM-writeboost could found “slow request”.
  - Logic block cache device didn’t response to BlueStore OSD write request in time.
  - Suggest deploy different SSD disks for Cache and BlueStore.

- SSD management consistency between BlueStore and Block Cache
  - Data in SSD is labeled as “dirty_data” in Bcache even all data deleted in Ceph.
  - BlueStore doesn’t support discard/TRIM currently(https://github.com/ceph/ceph/pull/14727)

- Logic disk recover problem after host restart
  - There is no file system on the logic block, so disk label miss/reorder after host restart (Bcache https://bugs.launchpad.net/curtin/+bug/1728742 https://github.com/koverstreet/bcache-tools/pull/1)
  - It takes long time to recover each logic disk when SSD cached many data(DM-writeboost)

Future direction suggestion

- Ceph BlueStore controls raw disk and has different allocators to manage raw disks.
- Linux block cache also controls raw disk allocation.
- There might be some inconsistency between BlueStore and Block cache, especially for SSD device.
- It would be better to let BlueStore overall controls raw disks. Moreover, BlueStore can control data priority to save to fast device.



Thank You!