Administration & Operations

Durable Deletes

AEROSPIKE

Shared RAM Memory Utilization – Enterprise Edition Only

- Linux Shared Memory (persists on aerospike service restart on a node)
- Stores Primary Index in order of namespaces declared in the config file.
- `$sudo ipcs –m`
  - Displays shared memory segments – two blocks per namespace.
  - 0xAE001... 2MB, 0xAE0011... 1GB
  - Further details at:
  - http://www.aerospike.com/docs/operations/manage/aerospike/fast_restart.html
- Stores data if data-in-index and single bin integer.
- If passes version validation, ‘start’ initiates FastStart – index is not rebuilt off SSD persistent store.

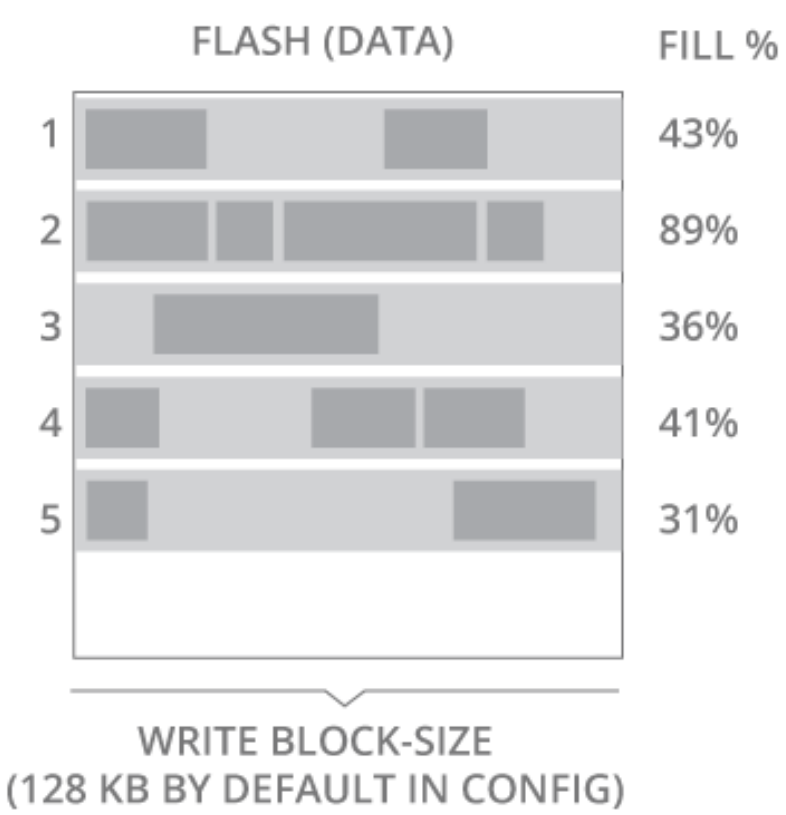
Process RAM Memory Utilization

- Community Edition:
  - Stores Primary Index (PI)
  - Stores data if data-in-index and single bin integer.
- Process Memory is allocated every time Aerospike server starts.
- => CE always coldstarts – ie it must rebuild PI from persistent store.
- Data-in-Memory – always stored on Process RAM.
- Secondary Index – always stored on Process RAM.
- Both for CE and EE, if Data-in-memory then persistent store (file on HDD typically) has to be scanned.
- Persistent Store when HDD, (file storage) it is typically used when using data-in-memory as a backup.

namespace – device config

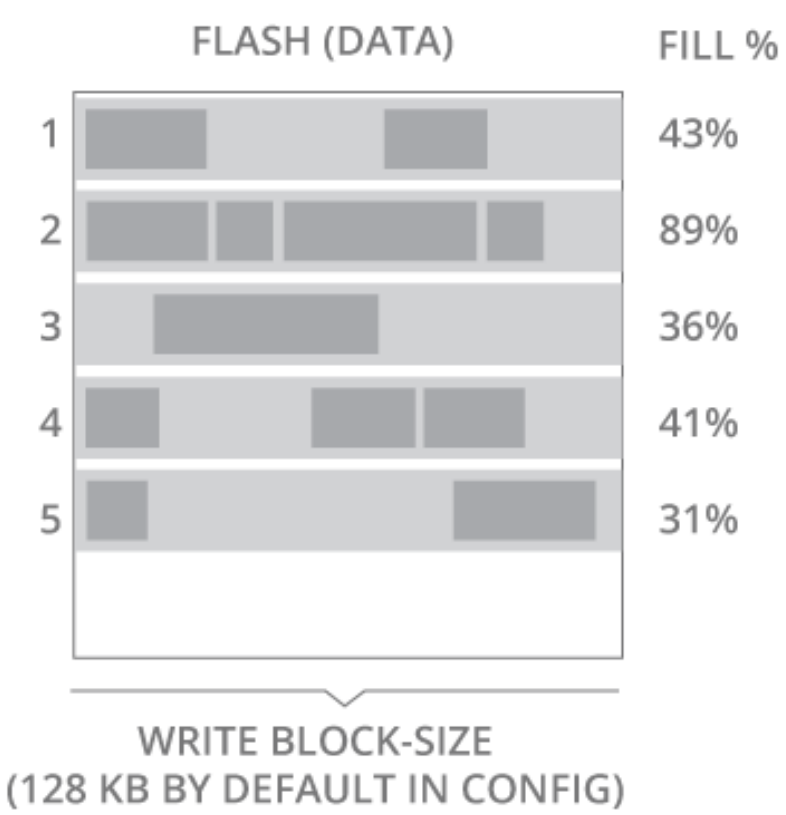
- If you want to coldstart the server but want to ignore the data on the persistent store, in namespace device configuration, use:
  - `cold-start-empty true [VERY DANGEROUS!!]`
- Caveat:
  - Once you start with this configuration, there is no guarantee that old deleted records will not come back if you disable it for subsequent restarts.
  - So once you invoke it, you have to always use it.
  - If server restarts on its own due to power failure, you will lose all current data.
  - Better – don't use it, wipe out your SSD (zeroize it using "dd" command)
  - For files, delete the data file at /opt/aerospike/data/...

Zombie Records – Case 1



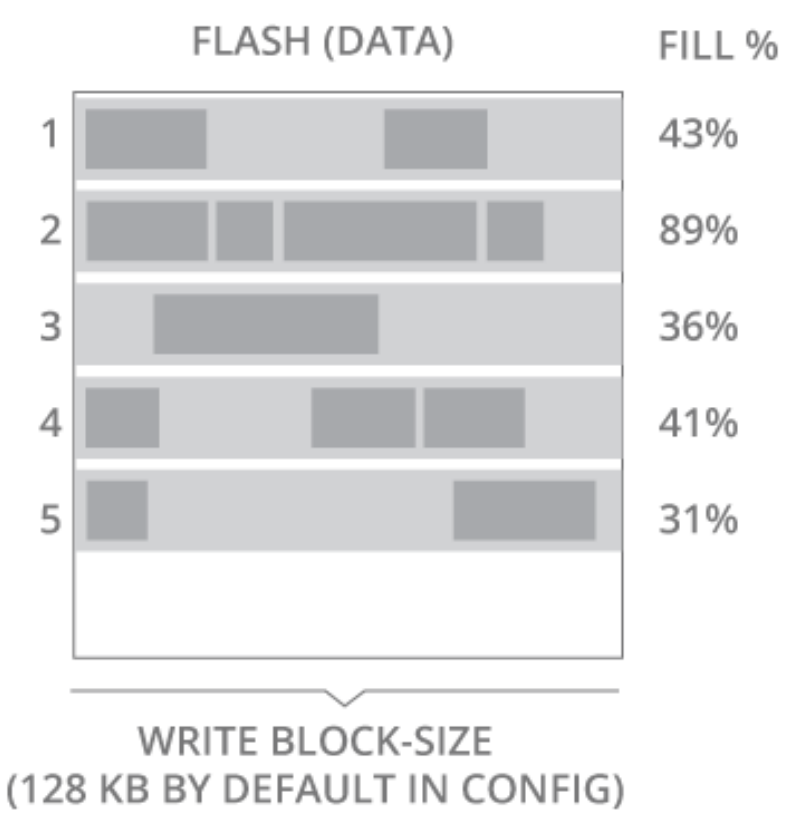
- When a record is deleted, its PI is deleted from RAM.
- Record on SSD is marked for defrag when defrag-lwm-pct is reached.
- Defrag puts the block in free queue.
- Even in free queue, till overwritten by new data, old record data is still there.
- If this node goes down, (cluster has deleted this record), node comes back up and rejoins the cluster in **coldstart** mode, this record will come back to life.

Zombie Records – Case 2



- Consider a valid record on this node, and this node goes down.
- Subsequently, this record is deleted by cluster.
- This node rejoins the cluster at a later time in FastRestart (finds the PI) or coldstart (rebuilds the PI) – either case deleted record will come back.
- Solution?
- Don’t use DELETE.
- Use initial TTL to expire records.

Zombie Records – Case 3 – TTL update



- Consider a valid record on this node.
- Initial TTL was 30 days.
- You change it to 10 sec, the record expires.
- Then unexpectedly, this node goes down.
- Now if you coldstart the node (CE or EE because node crashed, linux shared memory is gone), depending on which TTL version is scanned first, record may or may not come back. (Hint: If 30d TTL is scanned first, we will be fine – Why?)

Zombie Records - Solution

- Aerospike is optimized for performance if you only use initial TTL at time of record creation to expire the record.
- ie Don’t use DELETE or change the TTL
- If you must DELETE, use DURABLE_DELETE policy (Ver 3.10+)
- By default, DURABLE_DELETE is FALSE.
- Plan storage and RAM to account for Tombstones if using Durable Delete.
- Plan for performance hit due to Tomb-Raider cycles.

Zombie Records – Solution (Durable Delete)

- Durable Delete writes a Tombstone.
- Tombstone is Record with all bins null, just metadata + digest.
  - Metadata stores GENERATION and LAST UPDATE TIME.
  - TTL is set to “LIVE FOR EVER” (-1)
- Tombstone is deleted by Tomb-Raider when all below are true:
  - All previous versions of the record have been overwritten
  - Current time > Tombstone Last Update Time + Tombstone minimum life (1d default)
  - All Migrations are complete.
- Tombstones protect you from Zombie records for node restarts within Tombstone minimum life.

Starting And Stopping Aerospike Server

Controlling the server requires you to be root or have sudo privileges.

Start server

```
sudo service aerospike start
```

Coldstart server

```
sudo service aerospike coldstart
```

Check on status

```
sudo service aerospike status
```

Stop server

```
sudo service aerospike stop
```

Restart server

```
sudo service aerospike restart
```

Exercise

Setup a three or more node cluster using Aerospike Enterprise Edition.

Use:

```
namespace test {
    replication-factor 2
    memory-size 10M
    default-ttl 3d # 3 days

    storage-engine device {
    file /opt/aerospike/data/test.dat
    filesize 100M
    #cold-start-empty true
    data-in-memory false
    post-write-queue 8
    min-avail-pct 20
    }
}
```

Exercise

- Insert a record using AQL:

```
aql>SET KEY_SEND TRUE
aql> INSERT INTO test.demo (PK, foo, bar) VALUES ('key1', 123,
'abc')
```

- On AMC dashboard, see which node is the master.
- On Master: `sudo service aerospike stop`

Wait till migrations are complete.
Using AQL, delete the record on the remaining two node cluster.

```
aql>SET DURABLE_DELETE FALSE
aql>DELETE FROM test.demo WHERE PK = 'key1'
```

- FastStart (start) stopped node.
- Since we did not use durable deletes, record comes back?
- Repeat with coldstart.
- Stop all nodes. Delete /opt/aerospike/data/test.dat.
- Repeat with Durable_Delete true.