Exercise 16 – Write Path

In this exercise, you will:

- Understand the Apache Cassandra™ write path.

Apache Cassandra™ has an optimized write path. To understand how to use Apache Cassandra™, it can be very helpful to understand the Apache Cassandra™ write path. In this exercise we will see Apache Cassandra™ writing data to the file system.

Steps

1) Let's simplify our cluster a bit. Shut down all three nodes and delete the folders node1, node2 and node3.

2) Make a fresh new node by extracting it from the tarball by executing the following commands in the terminal from within the /home/ubuntu directory:

```
tar -xf dse-6.0.0-bin.tar.gz
mv dse-6.0.0 node
labwork/config_node
```

3) Now start this node.

```
/home/ubuntu/node/bin/dse cassandra
```

4) Check the status of the running node `/home/ubuntu/node/bin/nodetool status`.

   NOTE: If the status shows two nodes and the second is DN, follow the instructions on this link to remove the second node before proceeding: https://docs.datastax.com/en/opscenter/5.1/opsc/online_help/opscRemovingPackages_t.html

5) Investigate the commit-log directory.

```
ls -lh /home/ubuntu/node/data/commit-log
```

```
ubuntu@ds201-node1:~$ ls -lh /home/ubuntu/node/data/commit-log
total 68K
-rw-rw-r-- 1 ubuntu ubuntu 61K Apr 16 23:17 CommitLog-7-1523920550026.log
```

```
-rw-rw-r-- 1 ubuntu ubuntu  20 Apr 16 23:15 CommitLog-7-1523920550027.log
```

6) Let's put a watch on this directory to see how it changes as we write data to Apache Cassandra™. Open a second terminal and `ssh` to the remote machine.

7) In the new terminal, execute the following command:

   ```
   watch -n 1 -d "ls -lh /home/ubuntu/node/data/commit-log"
   ```

   NOTE: To exit the watch later, press CTRL-C

8) We will now use the `cassandra-stress` tool to write several thousand records to our node. Execute the following command in your original terminal:

   ```
   /home/ubuntu/node/resources/cassandra/tools/bin/cassandra-stress
   write no-warmup n=250000 -port native=9041 -rate threads=1
   ```

   Be sure your second terminal is also visible as `cassandra-stress` executes. `cassandra-stress` will write 250,000 rows to your node.

   There are a few things to watch out for while `cassandra-stress` inserts keys:

   - The total size will continue to increase.
   - The timestamp will change for the current segment being written.
   - You may get additional commit log files as well.

9) When `cassandra-stress` completes, terminate the watch by pressing `CTRL-C`.

10) Execute the following `nodetool` command:

   ```
   /home/ubuntu/node/bin/nodetool cfstats keyspace1.standard1
   ```

   `cassandra-stress` created the `keyspace1.standard1` table and populated its data. `cfstats` gives you column family stats. Column family is a deprecated term for a table.

   ```
   ubuntu@ds201-node1:~$ ./node/bin/nodetool cfstats
   keyspace1.standard1
   Total number of tables: 47
   ----------------
   Keyspace : keyspace1
           Read Count: 0
           Read Latency: NaN ms
           Write Count: 250000
   ```

```
        Write Latency: 0.04085058 ms
        Pending Flushes: 0
                Table: standard1
                SSTable count: 2
                Space used (live): 59810839
                Space used (total): 59810839
                Space used by snapshots (total): 0
                Off heap memory used (total): 310184
                SSTable Compression Ratio: -1.0
                Number of partitions (estimate): 252133
                Memtable cell count: 649
                Memtable data size: 181071
                Memtable off heap memory used: 0
                Memtable switch count: 5
                Local read count: 0
                Local read latency: NaN ms
                Local write count: 250000
                Local write latency: 0.034 ms
                Pending flushes: 0
                Percent repaired: 0.0
                Bytes repaired: 0.000KiB
                Bytes unrepaired: 54.450MiB
                Bytes pending repair: 0.000KiB
                Bloom filter false positives: 0
                Bloom filter false ratio: 0.00000
                Bloom filter space used: 310200
                Bloom filter off heap memory used: 310184
                Index summary off heap memory used: 0
                Compression metadata off heap memory used: 0
                Compacted partition minimum bytes: 180
                Compacted partition maximum bytes: 258
                Compacted partition mean bytes: 258
                Average live cells per slice (last five minutes): NaN
                Maximum live cells per slice (last five minutes): 0
                Average tombstones per slice (last five minutes): NaN
                Maximum tombstones per slice (last five minutes): 0
                Dropped Mutations: 0
                Failed Replication Count: null
```

Notice the "Write Count" matches the number of rows we told `cassandra-stress` to insert. `cfstats` also reports the number of SSTables, space used, and bloom filter statistics.

11) Note the Memtable statistics.

```
Memtable cell count: 649
Memtable data size: 181071
Memtable off heap memory used: 0
Memtable switch count: 5
```

12) Execute the following `nodetool` command which will flush the memtable contents to disk.

```
/home/ubuntu/node/resources/cassandra/bin/nodetool flush
```

13) Now check the table stats again by executing

```
/home/ubuntu/node/resources/cassandra/bin/nodetool cfstats
keyspace1.standard1
```

Note the memtable statistics zeroed out because we flushed the previous memtable to disk.

```
Memtable cell count: 0
Memtable data size: 0
Memtable off heap memory used: 0
Memtable switch count: 6
```

14) Another simple exercise you can do is shut down your node, delete the `logs/system.log` file, restart your node, then search for `CommitLog.java` in the new `logs/system.log` file. You may see lines reporting replays.

If there were no commit log segments found during startup, no replay needs to be done. If Apache Cassandra™ finds commit log files, it will replay the mutations in those files into memtables and then flush the memtables to disk.