Exercise 13 – Consistency Levels

In this exercise, you will:

- Understand the differences between consistency vs. replication factor.
- Understand the differences of consistency on a write vs. a read.

One of the challenges of distributed systems like Apache Cassandra™, is how to keep the data replicas consistent. Maintaining consistency requires balancing availability and partitioning. Fortunately, Apache Cassandra™ lets you tune this balancing to your needs.

In this exercise we will tinker with consistency levels using operations that read and write.

Steps

1) Ensure all three of your nodes are up. Then start `cqlsh` and switch to the `killrvideo` keyspace by executing the command below.

   ```
   USE killrvideo;
   ```

2) Now check the current consistency level by executing the CONSISTENCY command as we have it written out for you here:

   ```
   CONSISTENCY;
   ```

   Notice that consistency is ONE; meaning only one node must acknowledge a write on a write request, and only one node must return a result set to satisfy a read request.

3) Set your consistency level to TWO by executing the following command:

   ```
   CONSISTENCY TWO;
   ```

   NOTE: In this case, TWO is the same as ALL because our current replication settings store one replica per data center, and we have two data centers.

4) Retrieve the `cassandra` partition from the `videos_by_tag` table by executing the following command:

   ```
   SELECT *
   ```

```
FROM videos_by_tag
WHERE tag = 'cassandra';
```

The query succeeds.

5) Determine which nodes hold the replicas for the `cassandra` partition tag value in the `videos_by_tag` table by executing the following command at the terminal prompt:

```
/home/ubuntu/node1/resources/cassandra/bin/nodetool getendpoints
killrvideo videos_by_tag 'cassandra'
```

6) Bring one of the nodes down by executing:

```
/home/ubuntu/nodeX/resources/cassandra/bin/nodetool stopdaemon
```

where X is the node number. In choosing your nodes, pick the one that has another node with it in a rack. Note the node number correlates with the last value in the IP address listed in your terminal. Track which node it is that you brought down and also the node number that is still up.

7) Again, retrieve the `cassandra` partition by executing the following command in `cqlsh` (Note that you may have already executed the first two of these commands, but we include them in case you had to exit CQLSH. At any rate, it will not hurt to run them again):

> NOTE: By default, `cqlsh` connects to 127.0.0.1. If you ever bring down this node, you must tell `cqlsh` to connect to a different node by writing
>
> ```
> nodeX/bin/cqlsh 127.0.0.X
> ```
>
> where **X** is the number of the node you are connecting to. If you bring down node2, you also must specify the port to connect to as well by writing
>
> ```
> nodeX/bin/cqlsh 127.0.0.X 904X
> ```

```
USE killrvideo;

CONSISTENCY TWO;

SELECT *
FROM killrvideo.videos_by_tag
WHERE tag = 'cassandra';
```

The query will error out because you cannot achieve a consistency level of TWO since we took one of the needed nodes down.

8) Set the consistency level back to one:

```
CONSISTENCY ONE;
```

9) Now retry your query:

```
SELECT *
FROM killrvideo.videos_by_tag
WHERE tag = 'cassandra';
```

The query succeeds because we still have one replica node available.

10) Change the consistency level back to two:

```
CONSISTENCY TWO;
```

11) Let's try inserting a new row into the `cassandra` partition by executing the following command:

```
INSERT INTO killrvideo.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', '2016-2-8', uuid(), 'Me Lava Cassandra');
```

Notice the write failed because we still cannot achieve a consistency level of two.

12) Change the consistency level back to ONE:

```
CONSISTENCY ONE;
```

13) Now try the INSERT again:

```
INSERT INTO videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', '2016-2-8', uuid(), 'Me Lava Cassandra');
```

The write succeeds.

14) Now restart the node you took down. Wait for it to finish starting up before continuing.

15) Open `cqlsh` again and set your consistency level back to TWO:

```
CONSISTENCY TWO;
```

16) Now run the SELECT command again:

```
SELECT *
FROM killrvideo.videos_by_tag
WHERE tag = 'cassandra';
```

Notice the new record is present. This is due to read repair which will explore later.

17) Now, let's have a little fun. Change your consistency level back to ONE:

```
CONSISTENCY ONE;
```

18) Now bring down the other node that was up when we originally did our write. Wait for the node to terminate before continuing.

19) Start `cqlsh` again and re-execute the SELECT command.

```
SELECT *
FROM killrvideo.videos_by_tag
WHERE tag = 'cassandra';
```

Notice the query succeeds and the new record is still present. This is because of read repair, which we will explore later.

20) Update the record to a new title by executing the CQL command that follows. You will have to copy and paste the video_id from the result set above into the UPDATE's WHERE clause.

```
UPDATE killrvideo.videos_by_tag
SET title = 'Me LovEEEEEEEE Cassandra'
WHERE tag = 'cassandra' AND added_date = '2016-02-08' AND
video_id = paste_your_video_id;
```

21) Run the following SELECT command to ensure your update succeeded:

```
SELECT *
FROM killrvideo.videos_by_tag
WHERE tag = 'cassandra';
```

22) Now, bring down the node containing your replica. Keep track of which node this is. Wait for the node to terminate before continuing. Use either `dsetool` or `nodetool` to verify that both of your replica nodes are down (the nodes responsible for the `cassandra` tag).

23) Now bring up the other replica node (the one you didn't barely shut down). Wait for the node to start up before continuing.

24) Now execute the following SELECT command:

```
SELECT *
FROM killrvideo.videos_by_tag
WHERE tag = 'cassandra';
```

Your video title may be "Me Lava Cassandra" or "Me LovEEEEEEEE Cassandra" depending on which nodes acted as coordinator on the previous writes and whether they performed hinted handoffs. More on this to come.