

Flushes and Compactions

Objective: Learn how to manually flush and compact a table and view the results of each operation.

Successful outcome: You will manually flush and compact a table and view the results of each operation.

HDFS paths: `/apps/hbase/data` `/apps/hbase/data/WALs`

HBase tables: `sensor`

Now lets manually flush and compact a table and view the results of each operation. You will look at the data stored in the Write Ahead Log (WAL).

The table below has a column called HBase shell and another column called Terminal. Open a second Terminal window. In one terminal, run all of the commands in the HBase Shell column and in the other terminal run all of the commands shown in the Terminal column.

Step	HBase Shell	Terminal
1		<code>\$ hdfs dfs -ls -R /apps/hbase/data</code>
		You may need to do this: <code>\$ hdfs dfs -mkdir /apps/hbase/data/</code>
		This directory may already be there. Your

		directory may differ.
	<pre>> create 'sensor',NAME => 'cf', VERSIONS => 3</pre>	
2		<pre>\$ hdfs dfs -ls -R /apps/hbase/data/data/default/sensor</pre>
		<pre>\$ hdfs dfs -ls /apps/hbase/data/data/default/sensor/*/c</pre>
		No data has been added so no Region Store files exist in the column family directory
3	<pre>> put 'sensor', 'row1', 'cf:col', 'foo'</pre>	
4		<pre>\$ hdfs dfs -ls /apps/hbase/data/data/default/sensor/*/c</pre>
		The put is in the Region Memstore and has not yet been flushed to disk
5	<pre>> flush 'sensor'</pre>	
6		<pre>\$ hdfs dfs -ls /apps/hbase/data/data/default/sensor/*/c</pre>
		The flush has written the data in the Memstore to a Region Store file
7	<pre>> put 'sensor',</pre>	

	'row2', 'cf:col', 'bar'	
	> flush 'sensor'	
8		\$ hdfs dfs -ls /apps/hbase/data/data/default/sensor/*/c
		The second flush writes the contents of the Memstore into a new Region Store file
9		Read the files listed in step 8:
		\$ hbase hfile --printkv --file hdfs://localhost:8020/<full path from 8>
		Notice that you can view the Key Values from your previous put commands in the Region Store files
10	> major_compact 'sensor'	
11		\$ hdfs dfs -ls /apps/hbase/data/data/default/sensor/*/c
		\$ hbase hfile --printkv --file hdfs://localhost:8020/<path>
		A Major Compact will merge the smaller Region Store files into a single larger file. Notice that you can now see both Key Values in the new Region Store file

Compactions and Data

1. In the HBase shell, add another row with the following characteristics:

Table name: 'sensor'

Row key: 'row3'

Column Family: 'cf'

Column Descriptor: 'col' and value 'row 3 value'

2. Flush and run a major compaction on table `sensor`
3. List the file for the column family as shown in step 11 above.
4. Use the `hfile --printkv` command on the new file path to verify that row3
5. Delete the row that you just created.
6. Flush the delete changes that you just made.
7. Rerun the `hfile --printkv` command on the new file path to verify that row3 still exists even though the row was deleted.
8. Run a major compaction.

9. List the file for the column family as shown in step 11 above.

Use the `hfile --printkv` command on the new file path to verify that row3 does not exist after a compaction. During a major compaction deletions are cleaned up.

10. Add another row with the following characteristics:

```
Table name: 'sensor'  
Row key: 'row4'  
Column Family: 'cf'  
Column Descriptor: 'col' and value 'first time'
```

11. Update the row again with the following characteristics:

```
Table name: 'sensor'  
Row key: 'row4'  
Column Family: 'cf'  
Column Descriptor: 'col' and value 'second time'
```

12. Update the row again with the following characteristics:

```
Table name: 'sensor'  
Row key: 'row4'  
Column Family: 'cf'  
Column Descriptor: 'col' and value 'third time'
```

13. Update the row again with the following characteristics:

Table name: 'sensor'

Row key: 'row4'

Column Family: 'cf'

Column Descriptor: 'col' and value 'fourth time'

14. Flush the changes that you just made.

15. Run a major compaction.

16. List the file for the column family as shown in step 11 above.

17. Use the `hfile --printkv` command on the new file path to verify that row4 only has three versions.

Results

During a major compaction columns with too many versions are cleaned up. By default, all column families store one version, however in step 1 when you created table `sensor`, you specified that three versions be kept for column family cf. As you can see, the version with value “first time” (which was the oldest) has been dropped. Only the three latest version of the row are output.

<https://virtuant.github.io/hadoop-overview-spark-hwx/>Go Back

{% comment %}

Links for this lab

<https://www.ngdata.com/visualizing-hbase-flushes-and-compactions/>

<http://hadoop-hbase.blogspot.com/2014/07/about-hbase-flushes-and-compactions.html>

<http://www.dummies.com/programming/big-data/hadoop/compactions-in-hbase/>

<https://issues.apache.org/jira/browse/HBASE-14918>

Write Ahead Log

Unless bypassed, all interactions with HBase are placed in the Write Ahead Log (WAL). These interactions can be viewed using a utility bundled with HBase.

1. List the WAL files for the RegionServer:

```
hdfs dfs -ls /apps/hbase/data/WALs
```

2. Use the hbase hlog utility to see the contents of the WAL. The command format is:

```
hbase hlog hdfs://localhost:8020/<walpath>
```

For <walpath> use the path for the `.meta` file found in Step 1. An example is shown here:

```
hbase hlog
```

Results:

```
Sequence 426 from region 1588230740 in table hbase:meta at write timestamp: Fri Mar 06 20:37:36 PST 2016
```

```
Action:
```

```
row: tablesplit,,1425611550111.2323e2f770ad442a8757460690d6ee96\.
```

```
column: info:server
```

```
timestamp: Fri Mar 06 20:37:36 PST
```

```
2016 tag: []
```

```
Action:
```

```
row: tablesplit,,1425611550111.2323e2f770ad442a8757460690d6ee96\.
```

```
column: info:serverstartcode
```

```
timestamp: Fri Mar 06 20:37:36 PST
```


2016 tag: []
Action:
row: tablesplit,,1425611550111.2323e2f770ad442a87574
60690d6ee96\.
column: info:seqnumDuringOpen
timestamp: Fri Mar 06 20:37:36 PST
2016 tag: []
Sequence 427 from region 1588230740 in table hbase:m
eta at write
timestamp: Fri Mar 06 20:37:36 PST 2016
Action:
row: tablesplit,M1425611550111.0b82b222a4289cf8bb6b5
56011793d30.
...

Look through the changes that are recorded in the WAL.

Note: that the WAL files for the .meta. table end in '.meta', while user table WAL files do not have that extension.

```
{% endcomment %}
```