Big Data Infrastructures
Fall 2018

Lab 03 : HDFS + Hbase

Author : Thomas Schaller, Sylvain Julmy

Professor : Philippe Cudré-Mauroux

Task 1

We used the following command to copy the Weblogs to HDFS :

```
./bundle/hadoop/bin/hdfs dfs -copyFromLocal
     ./weblogs_hbase.txt
     /bdi_2018/bdi18_07/
```

Task 2

We used the following Java code to read the data from the Weblogs file and put it into the HBase Table.

```java
public class CreateHBaseTable {

    static Configuration conf = HBaseConfiguration.create();

    public static void main(String[] args) throws Exception {

        String[] months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
        ↪  "Sep", "Oct", "Nov", "Dec"};

        conf.set("hbase.zookeeper.quorum", "diufrm210.unifr.ch");
        conf.set("hbase.zookeeper.property.clientPort", "2181");

        Connection connection = ConnectionFactory.createConnection(conf);

        String tableName = "weblogs_bdi18_07";
        // Connect to Hbase and create the table
        try {
            Admin hAdmin = connection.getAdmin();
            HTableDescriptor hTableDesc = new HTableDescriptor(
                    TableName.valueOf(tableName));
            hTableDesc.addFamily(new HColumnDescriptor("Months"));
            hTableDesc.addFamily(new HColumnDescriptor("Statistics"));
            System.out.println("Creating Table...");
            hAdmin.createTable(hTableDesc);
            System.out.println("Table created Successfully...");

        } catch (Exception e) {
            e.printStackTrace();
        }

        System.out.println("Connection to the table");
        Table table = connection.getTable(TableName.valueOf(tableName));

        try {

            System.out.println("Insert the datas");
            // Reading the dataset from HDFS
            Path path = new Path("/bdi_2018/bdi18_07/weblogs_hbase.txt");
            FileSystem fileSystem = FileSystem.get(new Configuration());
            BufferedReader bufferedReader = new BufferedReader(new
            ↪  InputStreamReader(fileSystem.open(path)));

            String line = bufferedReader.readLine();
            while (line != null) {

                String[] elements = line.split("\t");
                assert elements.length == 13;

                String rowKey = elements[0];

                Put put = new Put(Bytes.toBytes(rowKey));

                int totalNumberOfVisit = 0;

                for (int i = 1; i < elements.length; i++) {
                    String key = months[i - 1];
                    String valueStr = elements[i];
                    int value = Integer.parseInt(valueStr);
                    if (value == 0)
                        continue;
                    totalNumberOfVisit += value;
                    put.addColumn(
                            Bytes.toBytes("Months"),
                            Bytes.toBytes(key),
                            Bytes.toBytes(valueStr)
                    );
                }

                put.addColumn(
                        Bytes.toBytes("Statistics"),
                        Bytes.toBytes("Active"),
                        Bytes.toBytes(
                                totalNumberOfVisit > 20 ?
                                        "1" :
                                        "0"
                        )
                );

                table.put(put);

                line = bufferedReader.readLine();
            }

            System.out.println("Done");
            table.close();

        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```

Task 3

The following sections contains the output of the specified command.

1)

Command : `scan 'weblogs_bdi18_07'`

Output :

```
hbase(main):014:0> scan 'weblogs_bdi18_07'
ROW                        COLUMN+CELL
 0.308.86.81|2012          column=Months:Jul, timestamp=1543329698909, value=1
 0.308.86.81|2012          column=Statistics:Active, timestamp=1543329698909, value=0
 0.32.48.676|2012          column=Months:Jan, timestamp=1543329698919, value=3

...

 88.88.38.655|2012         column=Statistics:Active, timestamp=1543329752426, value=0
 88.88.600.360|2012        column=Months:Jan, timestamp=1543329752427, value=2
 88.88.600.360|2012        column=Statistics:Active, timestamp=1543329752427, value=0
 88.88.618.331|2012        column=Months:May, timestamp=1543329752429, value=1
 88.88.618.331|2012        column=Statistics:Active, timestamp=1543329752429, value=0
 88.88.644.648|2012        column=Months:Jan, timestamp=1543329752431, value=1
 88.88.644.648|2012        column=Statistics:Active, timestamp=1543329752431, value=0
 88.88.67.600|2012         column=Months:Jul, timestamp=1543329752432, value=3
 88.88.67.600|2012         column=Statistics:Active, timestamp=1543329752432, value=0
 88.88.687.376|2012        column=Months:Jan, timestamp=1543329752434, value=2
 88.88.687.376|2012        column=Statistics:Active, timestamp=1543329752434, value=0
27300 row(s) in 7.2460 seconds
```

2)

Command :

```
scan 'weblogs_bdi18_07', {
  STARTROW => '0.32.85.668|2012',
  ENDROW => '01.660.68.623|2012'
}
```

Output :

```
hbase(main):023:0> scan 'weblogs_bdi18_07', {STARTROW => '0.32.85.668|2012', ENDROW => '01.660.68.623|2012'}
ROW                               COLUMN+CELL
 0.32.85.668|2012       column=Months:Jul, timestamp=1543329698922, value=8
 0.32.85.668|2012       column=Statistics:Active, timestamp=1543329698922, value=0
 0.45.305.7|2012        column=Months:Feb, timestamp=1543329698926, value=1
 0.45.305.7|2012        column=Months:Jan, timestamp=1543329698926, value=1
 0.45.305.7|2012        column=Statistics:Active, timestamp=1543329698926, value=0
 0.46.386.626|2011      column=Months:Nov, timestamp=1543329698928, value=1
 0.46.386.626|2011      column=Statistics:Active, timestamp=1543329698928, value=0
 0.48.322.75|2012       column=Months:Jul, timestamp=1543329698931, value=1
 0.48.322.75|2012       column=Statistics:Active, timestamp=1543329698931, value=0
 0.638.50.46|2011       column=Months:Dec, timestamp=1543329698934, value=8
 0.638.50.46|2011       column=Statistics:Active, timestamp=1543329698934, value=0
 0.87.36.333|2012       column=Months:Aug, timestamp=1543329698937, value=7
 0.87.36.333|2012       column=Statistics:Active, timestamp=1543329698937, value=0
6 row(s) in 0.0090 seconds
```

3)

Command : `count 'weblogs_bdi18_07'`

Output :

```
hbase(main):004:0> count 'weblogs_bdi18_07'
Current count: 1000, row: 11.638.80.681|2012
Current count: 2000, row: 14.676.84.33|2012
Current count: 3000, row: 18.614.66.380|2012
Current count: 4000, row: 322.05.67.601|2012
Current count: 5000, row: 323.55.374.668|2011
Current count: 6000, row: 325.83.602.85|2011
Current count: 7000, row: 328.327.620.3|2012
Current count: 8000, row: 362.4.40.321|2012
Current count: 9000, row: 366.387.680.320|2012
Current count: 10000, row: 41.388.661.660|2011
Current count: 11000, row: 44.81.54.615|2011
Current count: 12000, row: 48.681.648.08|2011
Current count: 13000, row: 52.682.638.604|2011
Current count: 14000, row: 55.30.58.687|2012
Current count: 15000, row: 57.68.658.31|2011
Current count: 16000, row: 606.41.1.88|2012
Current count: 17000, row: 630.630.322.65|2012
Current count: 18000, row: 638.38.386.658|2012
Current count: 19000, row: 651.05.680.613|2012
Current count: 20000, row: 658.624.85.64|2012
Current count: 21000, row: 668.302.304.308|2012
Current count: 22000, row: 680.686.17.85|2012
Current count: 23000, row: 682.674.56.58|2012
Current count: 24000, row: 687.624.84.684|2011
Current count: 25000, row: 80.331.62.07|2012
Current count: 26000, row: 85.610.688.8|2011
Current count: 27000, row: 88.630.610.80|2012
27300 row(s) in 2.9290 seconds

=> 27300
```

Task 4

Each following section present the code which perform the specified task. Each code is also available inside the zip archive.

1)

Task : Retrieve only the contents of the Columns: “Jan” and “Feb” from the row key: 06.305.307.336|2012.

Java code :

```java
// Retrieve only the contents of the Columns:
// Jan and Feb from the row key: 06.305.307.336|2012

Get get1 = new Get(Bytes.toBytes("06.305.307.336|2012"));
Result result1 = table.get(get1);

String janValue = Bytes.toString(result1.getValue(
    Bytes.toBytes("Months"),
    Bytes.toBytes("Jan")
));

String febValue = Bytes.toString(result1.getValue(
    Bytes.toBytes("Months"),
    Bytes.toBytes("Feb")
));

System.out.printf("Jan : %s, Feb : %s \n", janValue, febValue);

System.out.println("Part 1 done");
```

2)

Task : Create a new ip and year, and fill in the table with the same values as the row with key: 01.660.70.74|2012

Java code :

```java
//Create a new ip and year, and fill in the table with the same values as the row with key:
↪   01.660.70.74|2012

String newIp = "8.8.8.8";
String newYear = "1971";
String rowKey = newIp + "|" + newYear;

Put put = new Put(Bytes.toBytes(rowKey));

Get get2 = new Get(Bytes.toBytes("01.660.70.74|2012"));
Result result2 = table.get(get2);

```

```java
NavigableMap<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>> map = result2.getMap();

for (Map.Entry<byte[], NavigableMap<byte[], NavigableMap<Long, byte[]>>> familyEntry :
↪  map.entrySet()) {

    byte[] family = familyEntry.getKey();

    for (Map.Entry<byte[], NavigableMap<Long, byte[]>> colEntry : map.get(family).entrySet()) {

        byte[] qualifier = colEntry.getKey();

        for (Map.Entry<Long, byte[]> timestampedValue :
        ↪  map.get(family).get(qualifier).entrySet()) {

            byte[] value = timestampedValue.getValue();

            put.addColumn(family, qualifier, value);

        }
    }

}

table.put(put);

System.out.println("Part 2 done");
```

Here is a snapshot which proves that the new rowKey has been inserted with the correct values.

```
hbase(main):008:0> get 'weblogs_bdi18_07','01.660.70.74|2012'
COLUMN                            CELL
 Months:Jul                       timestamp=1543329698943, value=1
 Statistics:Active                timestamp=1543329698943, value=0
2 row(s) in 0.0100 seconds

hbase(main):009:0> get 'weblogs_bdi18_07','8.8.8.8|1971'
COLUMN                            CELL
 Months:Jul                       timestamp=1543333405817, value=1
 Statistics:Active                timestamp=1543333405817, value=0
2 row(s) in 0.0300 seconds
```

3)

Task : Delete the row with key: 88.88.324.601|2012

Java code :

```java
//Delete the row with key: 88.88.324.601|2012

Delete delete = new Delete(Bytes.toBytes("88.88.324.601|2012"));
```

```

table.delete(delete);

System.out.println("Part 3 done");
```

Here is the a snapshot of the two same query. We just run the code above in between them.

```
hbase(main):011:0> get 'weblogs_bdi18_07','88.88.324.601|2012'
COLUMN                           CELL
 Months:Feb                      timestamp=1543329752411, value=20
 Months:Jan                      timestamp=1543329752411, value=37
 Months:Sep                      timestamp=1543329752411, value=74
 Statistics:Active               timestamp=1543329752411, value=1
4 row(s) in 0.0120 seconds

hbase(main):012:0> get 'weblogs_bdi18_07','88.88.324.601|2012'
COLUMN                           CELL
0 row(s) in 0.0060 seconds
```