**Perform the earlier operation using Java program.**

**The program should be able to check for the presence of 'customer' table.**

**In case the table exists, it must be dropped and recreated.**

*PROGRAM:*

HBaseBulkLoad:

package newproj;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.hbase.HBaseConfiguration;

import org.apache.hadoop.hbase.HColumnDescriptor;

import org.apache.hadoop.hbase.HTableDescriptor;

import org.apache.hadoop.hbase.MasterNotRunningException;

import org.apache.hadoop.hbase.ZooKeeperConnectionException;

import org.apache.hadoop.hbase.client.HBaseAdmin;

import org.apache.hadoop.hbase.client.HTable;

import org.apache.hadoop.hbase.client.Put;

import org.apache.hadoop.hbase.io.ImmutableBytesWritable;

import org.apache.hadoop.hbase.mapreduce.HFileOutputFormat;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.client.Put;

import org.apache.hadoop.hbase.io.ImmutableBytesWritable;

import org.apache.hadoop.hbase.util.Bytes;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class HBaseBulkLoad {

@SuppressWarnings("deprecation")

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

Configuration conf = HBaseConfiguration.create();

checkCreate(conf);

String inputPath = args[0];

HTable table=new HTable(conf,args[2]);

conf.set("hbase.mapred.outputtable", args[2]);

Job job = new Job(conf,"HBase\_Bulk\_loader");

job.setMapOutputKeyClass(ImmutableBytesWritable.class);

job.setMapOutputValueClass(Put.class);

job.setSpeculativeExecution(false);

job.setReduceSpeculativeExecution(false);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(HFileOutputFormat.class);

job.setJarByClass(HBaseBulkLoad.class);

job.setMapperClass(newproj.BulkLoadMap.class);

FileInputFormat.setInputPaths(job, inputPath);

TextOutputFormat.setOutputPath(job, new Path(args[1]));

HFileOutputFormat.configureIncrementalLoad(job, table);

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

private static void checkCreate(Configuration conf) throws MasterNotRunningException, ZooKeeperConnectionException, IOException

{

HBaseAdmin admin = new HBaseAdmin(conf);

HTableDescriptor des=new HTableDescriptor(Bytes.toBytes("customer"));

des.addFamily(new HColumnDescriptor("details"));

if(admin.tableExists("customer")){

System.out.println("Table Already exists!");

admin.disableTable("customer");

System.out.println("Table: customer deleted");

}

admin.createTable(des);

System.out.println("Table: customer sucessfully created");

return;

}

}

=========================================================

BulkLoadMap:

package newproj;

import java.io.IOException;

import org.apache.hadoop.hbase.HTableDescriptor;

import org.apache.hadoop.hbase.client.HBaseAdmin;

import org.apache.hadoop.hbase.client.Put;

import org.apache.hadoop.hbase.io.ImmutableBytesWritable;

import org.apache.hadoop.hbase.util.Bytes;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

//import org.apache.hadoop.mapreduce.Mapper.Context;

public class BulkLoadMap extends Mapper<LongWritable, Text, ImmutableBytesWritable, Put> {

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

String line = value.toString();

String[]parts=line.split(",");

String rowKey = parts[0];

//The line is splitting the file records into parts wherever it is comma (,) separated, and the first column are considered as rowKey.

ImmutableBytesWritable HKey = new ImmutableBytesWritable(Bytes.toBytes(rowKey));

//Here the row key is first converted to Bytes as Hbase understand its data as Bytes, and also object is created as ImmutableBytesWriteable

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

//This will write the rowKey values into Hbase while creating an object.

//Here the fields of tables inside Hbase is are stated to be written

HPut.add(Bytes.toBytes("details"), Bytes.toBytes("name"), Bytes.toBytes(parts[1]));

HPut.add(Bytes.toBytes("details"), Bytes.toBytes("location"), Bytes.toBytes(parts[2]));

HPut.add(Bytes.toBytes("details"), Bytes.toBytes("age"), Bytes.toBytes(parts[3]));

context.write(HKey,HPut);

//first we are creating instance PUT with 1st field as row key,

}

}

==================================================

**OUTPUT:**

Executing the jar file and then creating HFILE .

Load the HFiles into hbase and check fir the tables.

[acadgild@localhost training]$ hadoop jar BKLoad.jar /customers.dat /output4 customer

[acadgild@localhost training]$ hadoop fs -ls /output4/details

/SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.

17/05/05 17:51:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 1 items

-rw-r--r-- 1 acadgild supergroup 1509 2017-05-05 17:50 /output4/details/509a5e74aa9743d4a5a422190e515f7d

[acadgild@localhost training]$ hadoop fs -cat /output4/details/509a5e74aa9743d4a5a422190e515f7d

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.

17/05/05 17:51:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

DATABLK\*����������@�1detailsage[؍��181detailslocation[؍��IND1detailsname[؍��Amit2detailsage[؍��202detailslocation[؍��PAK2detailsname[؍��Sumit3detailsage[؍��263detailslocation[؍��AUS3detailsname[؍��Rohit4detailsage[؍��244detailslocation[؍��UK4detailsname[؍��NamitBLMFBLK2

��������@)�(��B�KIDXROOT2($��������@E�1detailsage[؍���2Z�IDXROOT2@!�\_��FILEINF2����������@PBUF�

BLOOM\_FILTER\_TYPEROW

=

BULKLOAD\_SOURCE\_TASK%attempt\_1493966073243\_0005\_r\_000000\_0

‑

BULKLOAD\_TIMESTAMP[؍�?

­

DELETE\_FAMILY\_COUNT

EARLIEST\_PUT\_TS[؍��

"

EXCLUDE\_FROM\_MINOR\_COMPACTION

KEY\_VALUE\_VERSION

LAST\_BLOOM\_KEY4

MAJOR\_COMPACTION\_KEY�

­

MAX\_MEMSTORE\_TS\_KEY

-

TIMERANGE [؍��[؍��

hfile.AVG\_KEY\_LEN

hfile.AVG\_VALUE\_LEN

)

hfile.LASTKEY4detailsname[؍����BLMFMET2nj��������@�3org.apache.hadoop.hbase.KeyValue$RawBytesComparator�-1��3GTRABLK"$��$ �

(08

@HPZ.org.apache.hadoop.hbase.KeyValue$KeyComparator`[

[acadgild@localhost training]$ hadoop jar hbase-server-0.98.1-hadoop2.jar completebulkload /output4 customer

hbase(main):022:0> scan 'customer'

ROW COLUMN+CELL

1 column=details:age, timestamp=1493985043074, value=18

1 column=details:location, timestamp=1493985043074, value=IN

D

1 column=details:name, timestamp=1493985043074, value=Amit

2 column=details:age, timestamp=1493985043074, value=20

2 column=details:location, timestamp=1493985043074, value=PA

K

2 column=details:name, timestamp=1493985043074, value=Sumit

3 column=details:age, timestamp=1493985043074, value=26

3 column=details:location, timestamp=1493985043074, value=AU

S

3 column=details:name, timestamp=1493985043074, value=Rohit

4 column=details:age, timestamp=1493985043074, value=24

4 column=details:location, timestamp=1493985043074, value=UK

4 column=details:name, timestamp=1493985043074, value=Namit

4 row(s) in 4.7490 seconds