# HBase

## Region:

The basic unit of scalability and load balancing in HBase is called a region. Regions are essentially contiguous range of rows stored together. They are dynamically split by the system when they become too large. Alternatevely, they may also be merged to reduce their numbers and required storage files.

## Export/Import data from HBase:

### Exporting data from HBase to HDFS.

Follow below steps:

### Create the /tmp Directory

Create the /tmp directory and set permissions:

http://www.cloudera.com/content/cloudera-content/cloudera-docs/CDH4/4.6.0/static/important.jpg  **Important**:

If you do not create /tmp properly, with the right permissions as shown below, you may have problems with CDH components later. Specifically, if you don't create /tmp yourself, another process may create it automatically with restrictive permissions that will prevent your other applications from using it.

Create the /tmp directory after HDFS is up and running, and set its permissions to 1777 (drwxrwxrwt), as follows:

$ sudo -u hdfs hadoop fs -mkdir /tmp

$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp

### Create the MapReduce system directories:

sudo -u hdfs hadoop fs -mkdir -p /var/lib/hadoop-hdfs/cache/mapred/mapred/staging

sudo -u hdfs hadoop fs -chmod 1777 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging

sudo -u hdfs hadoop fs -chown -R mapred /var/lib/hadoop-hdfs/cache/mapred

### Verify the HDFS File Structure

$ sudo -u hdfs hadoop fs -ls -R /

You should see:

drwxrwxrwt - hdfs supergroup 0 2012-04-19 15:14 /tmp

drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var

drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib

drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib/hadoop-hdfs

drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib/hadoop-hdfs/cache

drwxr-xr-x - mapred supergroup 0 2012-04-19 15:19 /var/lib/hadoop-hdfs/cache/mapred

drwxr-xr-x - mapred supergroup 0 2012-04-19 15:29 /var/lib/hadoop-hdfs/cache/mapred/mapred

drwxrwxrwt - mapred supergroup 0 2012-04-19 15:33 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging

### Start MapReduce

***for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-\*` ; do sudo service $x start ; done***

To verify services have started, you can check the web console. The JobTracker provides a web console http://localhost:50030/ for viewing and running completed and failed jobs with logs.

From HBase home directory i.e ***/usr/lib/hbase*** execute the following command.

***$hadoop jar hbase.jar export DeviceData /tmp/export***

### Importing data from HDFS to HBase table.

Create table in hbase like

Hbase> create ‘TableImport’, ‘dcf’

Then execute the below command from HBase home.

***$hadoop jar hbase.jar import ImportTable /tmp/export***

HBase Code

import java.io.IOException;

//

// Sample Hbase table creation

// by Anders Brownworth

// http://www.anders.com/

import org.apache.hadoop.fs.\*;

import org.apache.hadoop.conf.\*;

import org.apache.hadoop.hbase.\*;

import org.apache.hadoop.hbase.io.\*;

import org.apache.hadoop.hbase.client.\*;

public class CreateTable {

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

System.out.println( "starting..." );

System.out.println( "getting config..." );

HBaseConfiguration hc = new HBaseConfiguration( new Configuration( ) );

HTableDescriptor ht = new HTableDescriptor( "cdrs" );

// we'll use the call start time as a row key

ht.addFamily( new HColumnDescriptor( "number:" ) ); // from and to

ht.addFamily( new HColumnDescriptor( "company:" ) );

ht.addFamily( new HColumnDescriptor( "time:" ) ); // billing seconds

ht.addFamily( new HColumnDescriptor( "location:" ) ); // rate center city 1, state 1 and city 2, state 2

System.out.println( "connecting..." );

HBaseAdmin hba = new HBaseAdmin( hc );

System.out.println( "creating table..." );

hba.createTable( ht );

System.out.println( "done!" );

}

}

<http://diveintodata.org/2009/11/27/how-to-make-a-table-in-hbase-for-beginners/>

good tutorial, but how do you set the max column version using HColumnDescriptor, I have created an object of HColumnDescriptor and called the setMaxVersions(100) but the column version is still set to three. Below is my code.

public static void createHBaseTable(String tablename, String tables, Configuration conf) throws IOException {  
HTableDescriptor htd = new HTableDescriptor(tablename);  
for(String family : tables.split(“,”)){  
HColumnDescriptor hcd = new HColumnDescriptor(Bytes.toBytes(family));  
hcd.setMaxVersions(100);  
htd.addFamily(hcd);  
}  
HBaseAdmin admin = new HBaseAdmin(conf);  
if(admin.tableExists(tablename)) {  
admin.disableTable(tablename);  
admin.deleteTable(tablename);  
}

admin.createTable(htd);  
}

<http://archanaschangale.wordpress.com/2013/09/11/connecting-hbase-using-java/>

<http://bestlovejava.blogspot.in/2013/07/hbase-java-simple-example.html>

Spring HBase

<https://github.com/spring-projects/spring-data-book/tree/master/hadoop/hbase>

<http://docs.spring.io/spring-hadoop/docs/1.0.1.RC1/reference/html/hbase.html>

### Apache HBase in Standalone Mode and Connection using Java from Remote Machine

#### **Steps Performed :**

1. HBase is installed in the Machine with IP 10.97.136.138. With “root” as user and password is “root”.
2. Installed directory is: $/root/Softwares/hbase-0.98.9-hadoop2
3. Install HBase in Standalone Mode as defined in below URL. <http://hbase.apache.org/book/quickstart.html>
4. Edit the /etc/hosts file of the Machine as shown below.  
   *10.97.136.138 IN247itself*

*127.0.0.1 localhost*

*localhost IN247itself*

1. Edited the hbase-site .xml , present in the “conf” folder of the Installed “HBase” .  
     
   *<configuration>*

*<property>*

*<name>hbase.rootdir</name>*

*<value>file:////root/hbasedata/hbase</value>*

*</property>*

*<property>*

*<name>hase.zookeeper.quorum</name>*

*<value>10.97.136.138</value>*

*</property>*

*<property>*

*<name>hbase.zookeeper.property.clientPort</name>*

*<value>2181</value>*

*</property>*

*<property>*

*<name>hbase.zookeeper.property.dataDir</name>*

*<value>/root/hbasedata/zookeeper</value>*

*</property>*

*</configuration>*

1. Edit conf/hbase-env.sh as below. Add the below line only.  
   *export JAVA\_HOME=/usr*
2. To connect to the installed HBase from Windows Machine using java below steps were performed.
3. Ping command should work i.e ping 10.97.136.138 or ping IN347itself should work.
4. Edit hosts file under windows.
5. Hosts file present at the location “C:\Windows\System32\drivers\etc\hosts.
6. Add the blow line to the file  
   *10.97.136.138 IN247itself*
7. On Linux machine on which “HBase” is installed add the below commands  
     
   *$iptables –A INPUT –s 172.16.52.143 –j ACCEPT  
   $iptables –A INPUT –p tcp –dport 60010 –j ACCEPT  
   $/sbin/service iptables save*

*$service iptables restart*

1. 172.16.52.143 is IP of the Windows Machine.
2. Refer the below URL for More details on the above  
   <http://wiki.centos.org/HowTos/Network/IPTables>
3. Dependencies to be added to the POM file of the Java project.

*<dependency>*

*<groupId>org.apache.hbase</groupId>*

*<artifactId>hbase-client</artifactId>*

*<version>0.98.5-hadoop2</version>*

*</dependency>*

1. Installed version of HBase on server is “0.98.9-hadoop2”

Other to update ur personal data

https://in.linkedin.com/in/sreejithrpillai

The deliverables of the project was to retrieve business requirement from client's external feeds such as to identify the http response of client's website, most used telecom offers, analyzing session time of each user and it's different web page navigation, US zone wise data gathering on different offers, identifying different store locator and Ip Locator based on latitude and longitude from third party, .  
  
-Set up HDP Hadoop YARN 2.4.0.2.1, HDP Hbase 0.98.0.2.1, Zookeeper 3.4.5.2.1 on AWS EC2 instances. Hands on experience with installing Hadoop with Yarn, configuring Hbase and zookeeper and tweaking hbase regions configurations so as to achieve high performance.  
-Used different data ingestion adapters to dump data into hbase aiming high performance such as Bulk load, ImportTSV, and writing Map reduce code to read flat files to ingest data to optimize performance.  
-Designed complex row keys for hbase schema so as to align with the business requirement which resulted into high performace retrieval of data from hbase.  
-Analyzing data on hbase using filters and scanning on rowkeys to perform scan on hbase regions.  
-Exposing the end result through REST web service connecting web service and Hbase.