BigData Concepts

29 December 2023 08:35

HBase

- Introduction to HBase

  Lecture
- Overview of HBase
- HBase architecture
- Installation

The HBaseAdmin and HBase Security

Lecture

- Various Operations on Tables
- HBase general command and shell,
- java client API for HBase
- Admin API
- CRUD operations
- Client API
- HBase – Scan, Count and Truncate
- HBase Security

Lab-Assignment:

- Run the Hbase shell commands
- Run the HBase using Java client

Introduction to HBase

Hadoop -->

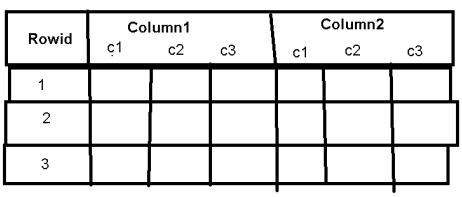
HDFS

- This is a distributed file systems suitable for storing large files
- Does not support fast individual record lookup
- There is no concept of batch processing --> it provides high latency batch processing
- Only sequential access to data is allowed

HBase

- Database built on top of HDFS
- HBase provides fast lookup for larger tables
- Random access --> It provides low latency access to single rows from all records (eg., millions or billions of records)
- It internally uses Hash tables and provides random access, and it stores the data in indexed HDFS files for faster lookups

Storing Data in HBase

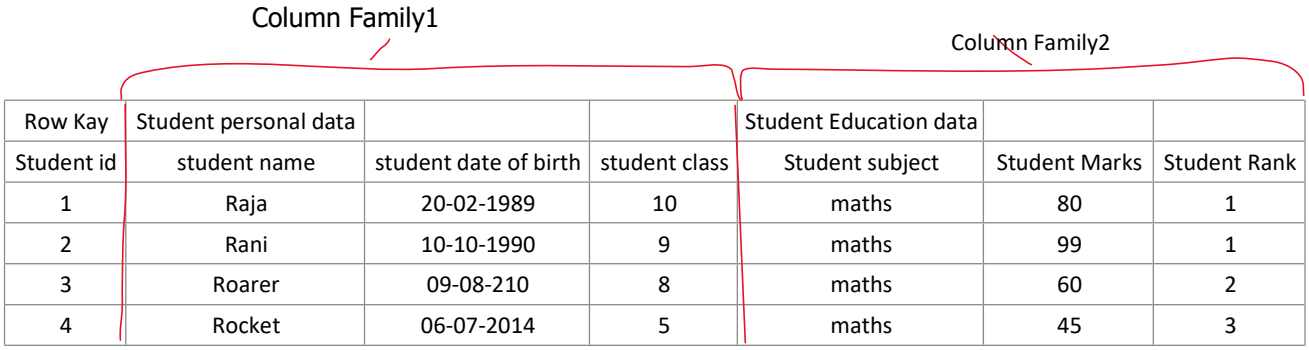
- It is column-oriented database and the tables in it are stored by row.
- The tables schema defines only column families, which are the key value pairs.
- 

- A table have multiple column families and each column family (in the above ex: column1, column2 …)can have any number of columns (c1, c2, c3)
- Subsequent column values are stored contiguously on the disk.

- Each cell value of the table has a timestamp

Column-oriented database:-

- It is suitable for online Analytical processing (OLAP)
- These are designed for huge tables



Row Kay	Student personal data			Student Education data		
Student id	student name	student date of birth	student class	Student subject	Student Marks	Student Rank
1	Raja	20-02-1989	10	maths	80	1
2	Rani	10-10-1990	9	maths	99	1
3	Roarer	09-08-210	8	maths	60	2
4	Rocket	06-07-2014	5	maths	45	3

- Tables is a collection of rows
- Row is a collections of column families
- Column family is a collection of columns
- Column is a collection of key value pairs

HBase comparison with RDBMS

RDBMS

1. It is governed by its schema - describes the whole structure of tables
2. It is thin and built for small tables - hard to scale
3. Transactional
4. Normalized data
5. This is very good for structured data

HBASE

1. It is schema-less, it doesn't have the concept of fixed columns schema; but we define only column families.
2. It is built for wide tables and we can scale horizontally
3. It is a not a transaction oriented (no transactions)
4. De-normalized data
5. Good for semi-structured as well as structured data

Hbase Features

- Hbase is linearly scalable
- Supports automatic failure handling (Zookeeper along with Hadoop clusters)
- It provide consistent read and write access
- Intergates with Hadoop, both as a source and destination
- JAVA API is available to access from the client
- Provides data replications across clusters

HBASE Installation

$ cd
$ mkdir HBase
$ cd Hbase
$ wget https://dlcdn.apache.org/hbase/2.4.17/hbase-2.4.17-bin.tar.gz
$ tar -zxvf hbase-2.4.17-bin.tar.gz
$ su
# mv hbase-2.4.17 /usr/local/hbase
# chown -R hadoop.hadoop /usr/local/hbase

nano /etc/bash.bashrc (update in all the nodes)

#HBASE CONFIGS
export HBASE_HOME="/usr/local/hbase"
export PATH="$HBASE_HOME/bin:$PATH"
<save and exit>

nano /usr/local/hbase/conf/hbase-enf.sh
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HBASE_PID_DIR=/var/hbase/pids
export HBASE_MANAGES_ZK=true
export HBASE_DISABLE_HADOOP_CLASSPATH_LOOKUP="true"
<save and exit>

# source /etc/bash.bashrc
# exit
$ source /etc/bash.bashrc
$ hbase version

```
hadoop@mainserver1:~/HBase$ hbase version
HBase 2.4.17
Source code repository git://e0b8ecc3178f/home/taklwu/hbase-rm/output/hbase revision=7fd096f39b4284da9a71
da3ce67c48d259ffa79a
Compiled by taklwu on Fri Mar 31 18:10:45 UTC 2023
From source with checksum 0e34884c8e1d6e46ba560b6a824d684ba4dd42b8f34a4318e57510627a88e3e41e34bdb8f06da56
a46d8065f331aa349ad58755cf6db5d75a5742f4174ccf81d
hadoop@mainserver1:~/HBase$
```

On master node only

$ cd /usr/local/hbase/conf
$ nano hbase-site.xml

```
<configuration>
 <property>
   <name>hbase.rootdir</name>
   <value>hdfs://mainserver1:9000/hbase</value>
 </property>

 <property>
   <name>hbase.cluster.distributed</name>
   <value>true</value>
 </property>

 <property>
   <name>hbase.tmp.dir</name>
   <value>./tmp</value>
 </property>
 <property>
   <name>hbase.unsafe.stream.capability.enforce</name>
   <value>false</value>
 </property>
 <property>
   <name>hbase.zookeeper.property.dataDir</name>
   <value>hdfs://mainserver1:9000/zookeeper</value>
 </property>
<property>
   <name>hbase.zookeeper.quorum</name>
   <value>mainserver1, slave1, slave2</value>
 </property>
 <property>
   <name>hbase.zookeeper.property.clientPort</name>
   <value>2181</value>
 </property>
</configuration>
```

<save and exit>

On slave nodes only
$ nano /usr/local/hbase/conf/hbase-site.xml

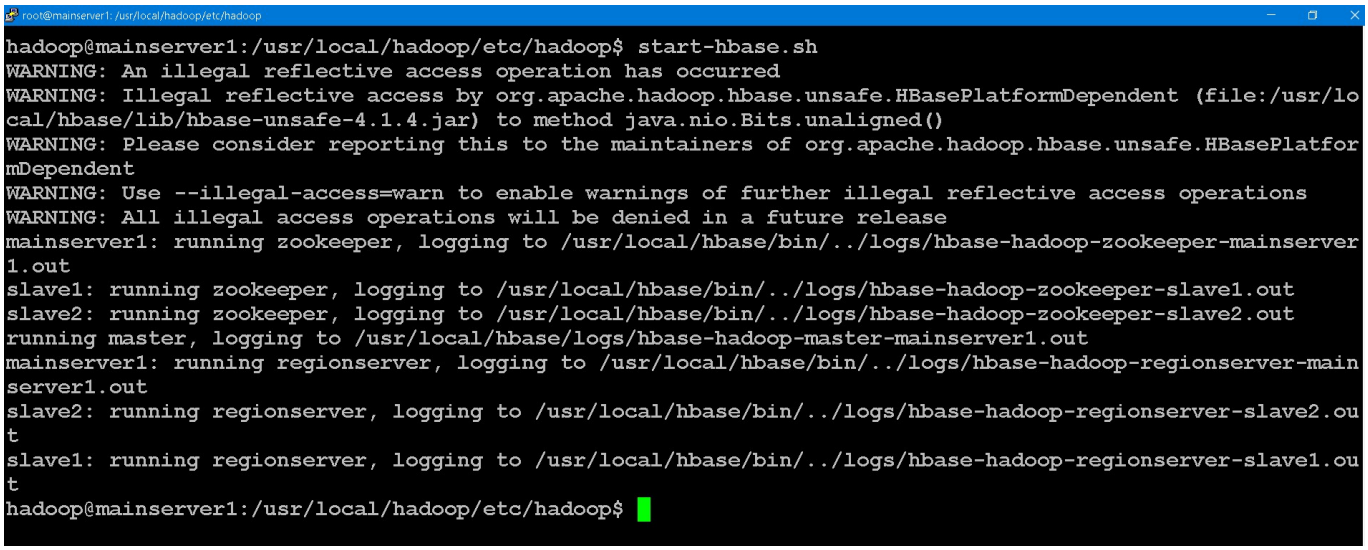
```
<configuraton>
<property>
   <name>hbase.rootdir</name>
   <value>hdfs://mainserver1:9000/hbase</value>
 </property>
<property>
 <name>hbase.cluster.distributed</name>
 <value>true</value>
 </property>
</configuration>
```

<save and exit>

Do this only on master node
# nano regionservers
$ cat regionservers
mainserver1
slave1
slave2
<save and exit>

Do this on master nodes
cd /usr/local/hadoop/etc/hadoop/
nano workers
mainserver1
slave1
slave2
<save and exit>

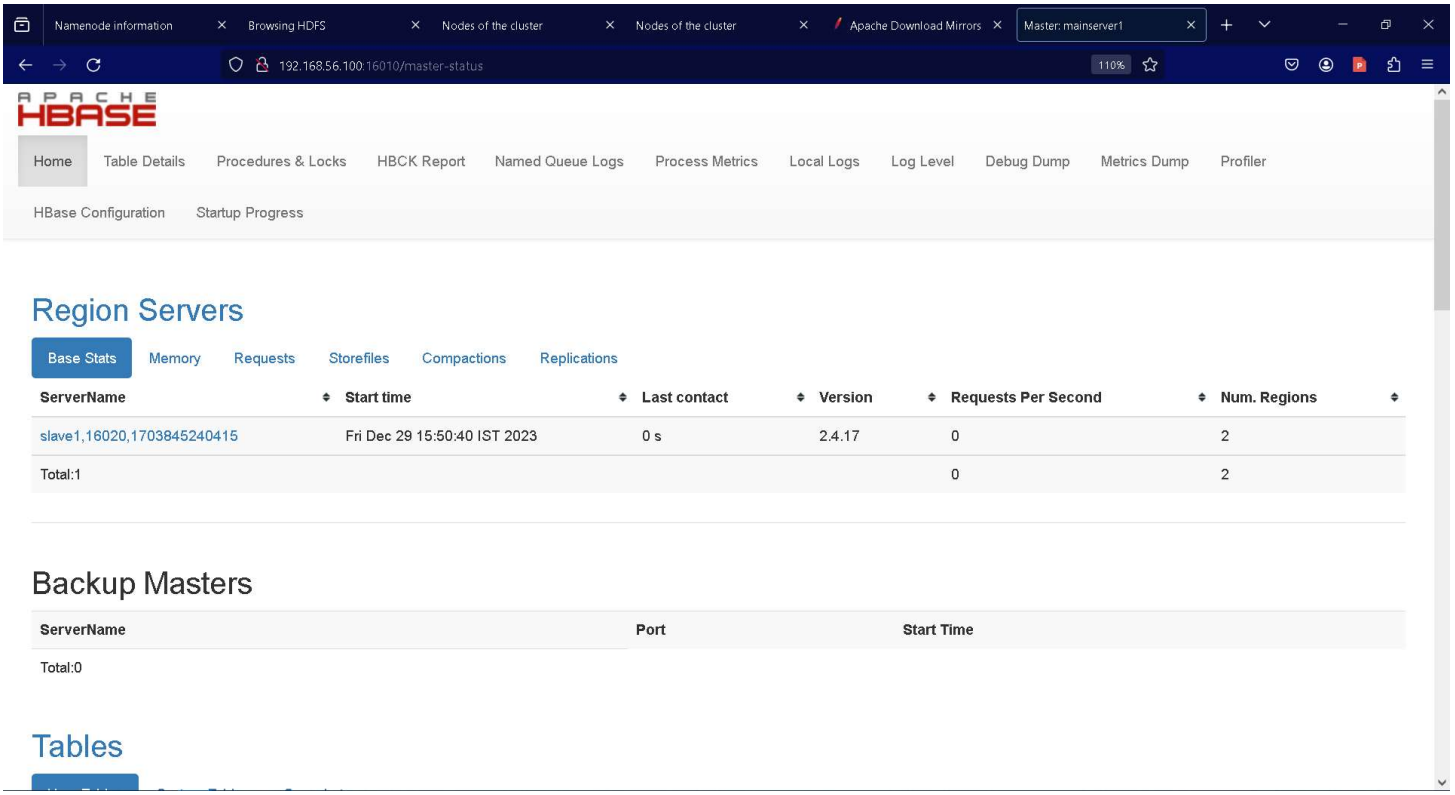
Execute on all the nodes
$ su -
# chmod 777 /var
 # chmod 777 /usr/local/hbase
 # chmod 777 /usr/local/hbase/
 # chmod 777 /usr/local/hbase/*
 # chmod 777 /var/
 # chmod 777 /var/*



```
hadoop@mainserver1:/usr/local/hadoop/etc/hadoop$ jps
7233 HRegionServer
1427 NameNode
9208 Jps
1832 ResourceManager
7000 HMaster
1660 SecondaryNameNode
6909 HQuorumPeer
hadoop@mainserver1:/usr/local/hadoop/etc/hadoop$
```

```
hadoop@slave2:~$ jps
4481 Jps
1425 NodeManager
3206 HQuorumPeer
3401 HRegionServer
1243 DataNode
hadoop@slave2:~$
```

```
hadoop@slave2:~$ jps
4481 Jps
1425 NodeManager
3206 HQuorumPeer
3401 HRegionServer
1243 DataNode
hadoop@slave2:~$
```



########## ********** Verified HBASE Working Fine ********** ##########