Columnar Databases: Cassandra

# Set up

Set up Python 2.7 and enable it in current shell.

$ sudo yum install -y centos-release-scl

$ sudo yum install -y python27

$ scl enable python27 bash

Download Cassandra tarball.

$ sudo wget http://apache.lauf-forum.at/cassandra/2.2.14/apache-cassandra-2.2.14-bin.tar.gz

Untar the file.

$ sudo tar -xvf apache-cassandra-2.2.14-bin.tar.gz

Add the bin subdirectory into $PATH.

$ PATH=$PATH:/home/cloudera/apache-cassandra-2.2.14/bin

Cd /opt/

sudo wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "https://download.oracle.com/otn-pub/java/jdk/8u201-b09/42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.tar.gz"

sudo tar xzf jdk-8u201-linux-x64.tar.gz

cd jdk1.8.0\_201/

sudo alternatives --install /usr/bin/java java /opt/jdk1.8.0\_201/bin/java 2

sudo alternatives --config java a daemon. There will be much output, shouldn’t be any errors.

$ sudo cassandra

Verify that it is running.

$ nodetool status

Output should be like this.

Datacenter: datacenter1

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

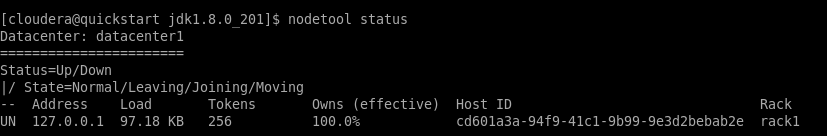
Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

-- Address Load Tokens Owns (effective) Host ID Rack

UN 127.0.0.1 169.05 KiB 256 100.0% ac401622-b894-47a9-a31b-e1f364adeaff rack1

Copy your own output here:



# Prepare the data

Download stock exchange dataset

$ wget https://s3.amazonaws.com/hw-sandbox/tutorial1/NYSE-2000-2001.tsv.gz -O - | gzip -d > NYSE-2000-2001.tsv

Cassandra uses an SQL dialect called CQL that allows to the users familiar with SQL to start operating in a very short amount of time. The Cassandra equivalent of a database as outermost grouping unit for data is called keyspace.

Open the cqlsh.

$ cqlsh

Create new keyspace.

cqlsh> CREATE KEYSPACE stock\_exchanges WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1};

Create a new table with partition key of (exchange, stock\_symbol).

cqlsh> CREATE TABLE stock\_exchanges.nyse

(exchange text,

stock\_symbol text,

date text,

stock\_price\_open float,

stock\_price\_high float,

stock\_price\_low float,

stock\_price\_close float,

stock\_volume float,

stock\_price\_adj\_close float,

PRIMARY KEY ((exchange, stock\_symbol), date));

# Import the data

Import it using cqlsh.

cqlsh> COPY stock\_exchanges.nyse (exchange , stock\_symbol , date, stock\_price\_open, stock\_price\_high,

stock\_price\_low, stock\_price\_close, stock\_volume, stock\_price\_adj\_close)

FROM 'NYSE-2000-2001.tsv'

WITH DELIMITER='\t' AND HEADER = 'true';

Copy here the output of the command:

... stock\_price\_low, stock\_price\_close, stock\_volume, stock\_price\_adj\_close)

... FROM 'NYSE-2000-2001.tsv'

... WITH DELIMITER='\t' AND HEADER = 'true';

Using 1 child processes

Starting copy of stock\_exchanges.nyse with columns [exchange, stock\_symbol, date, stock\_price\_open, stock\_price\_high, stock\_price\_low, stock\_price\_close, stock\_volume, stock\_price\_adj\_close].

Processed: 812989 rows; Rate: 2255 rows/s; Avg. rate: 4207 rows/s

812989 rows imported from 1 files in 3 minutes and 13.245 seconds (0 skipped).

# Query the data

Select the first five opening prices belonging to a specific ticker from the cqlsh.

cqlsh> SELECT date, stock\_price\_open FROM stock\_exchanges.nyse WHERE exchange = 'NYSE' AND stock\_symbol = 'ASP'

LIMIT 5;

Copy and paste text output or screenshot of at least first 5 lines here:

... LIMIT 5;

date | stock\_price\_open

------------+------------------

2000-01-03 | 11.19

2000-01-04 | 11.19

2000-01-05 | 11.13

2000-01-06 | 11.19

2000-01-07 | 11.13

(5 rows)

