Apache Hadoop

I. Introduction

This project is to calculate the relative frequency of co-occurrence of products that were purchased by customers. Example: The output's result can help in predicting the product that a custom would buy after buying a product.

II. Setup

2.0 **Prerequisites**

- 64bit host that support virtualization
- Minimum of 4GB of RAM available

2.1 **Virtual Box**

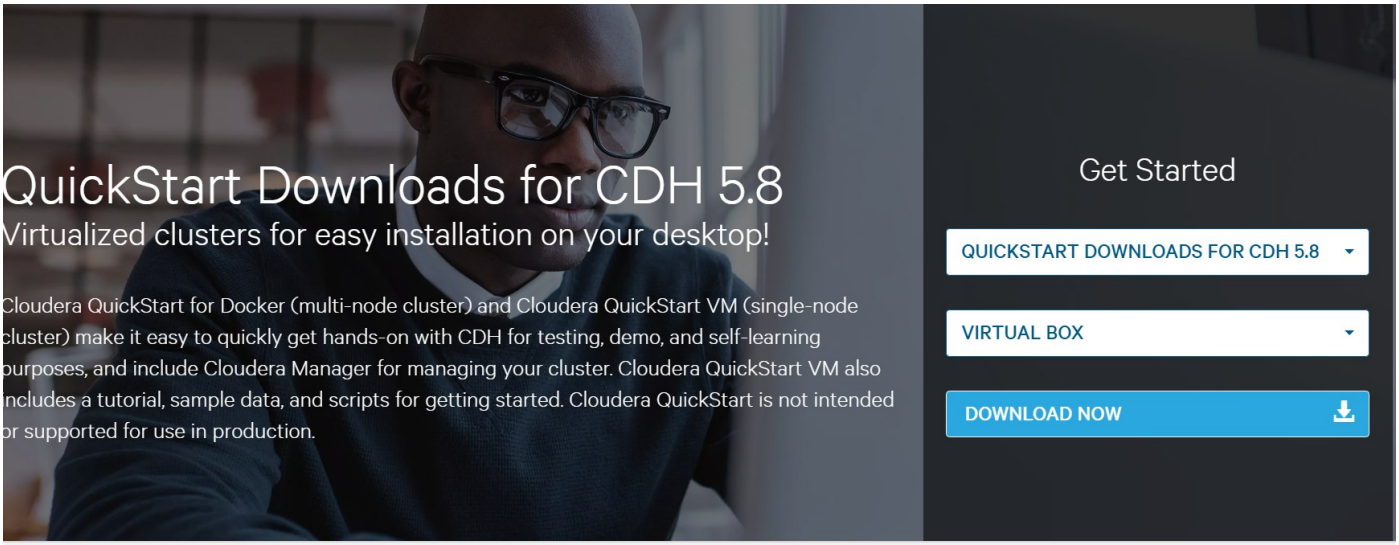
Dowanload the latest Oracle VirtualBox from the following link
https://www.virtualbox.org/wiki/Downloads
(At the time of making this document the latest version is *VirtualBox-5.1.6-110634*) then install the program

2.2 **Cloudera Quickstart**

This project, we are going to use Hadoop 2 in Cloudera Quickstart.
- Download Cloudera QuickStart for Hadoop from the following
http://www.cloudera.com/downloads/quickstart_vms/5-8.html

- Select the platform that you want to use (Here we are using Virtual Box)
- Click on DOWNLOAD NOW

– Sign in or fill out the product interest form to continue
– After the download completed, unzip the file, open the unzipped directory and click on the ovf file extension (eg: cloudera-quickstart-vm-5.8.0-0-virtualbox.ovf) this will taking care of setup the cloudera for virtual box.



III. Create Eclipse Project

Open Virtual Box, then start the cloudera quickstart. Wait till the the cloudera finished loading up.

Open Eclipse--> File --> New--> Java Project.

- Type project name (eg: HadoopProject) then click Finish

1. Add hadoop library for Elcipse

Click on created project, right click → Build Path → Config BuildPath

Select Libraries → Add External JARs → Goto path File System/usr/lib/hadoop

select all the jars files from there and the jar files in directories

/usr/lib/hadoop/lib and */usr/lib/hadoop/client-0.20*

Source code of the project is mentioned in section: IV. Generating jar and Testing

IV. Generating jar and Testings

Extract file from the folder hadoop.zip

input/	contains sample text file
partone/	class files
src/	source code (java files)
generateJar.sh	shell script for generating jar
runpair.sh	shell script for running pair approach
runstripe.sh	shell script for running stripe approach
runhybrid.sh	shell script for running hybrid approach
Result.txt	output result from shell scripts run for the 3 approac

4.1 Generating jar file

Go to unzipped folder (hadoop) right click → Open in Terminal

type: ./generateJar.sh

This should generate a jar class name hadoop1

4.2 Running Script and Result

input/

record.txt

This sample text file contains the following content

Mary 34 56 29 12 34 56 92 29 34 12

Kelly 92 29 12 34 79 29 56 12 34 18

Note:

- we are going to use this file for testing the 3 approaches: Pair, Stripe and Hybrid approaches
- number of reducer is set to 4 for all approaches

4.2 Pair

Go to unzipped folder (hadoop) right click → Open in Terminal

type: ./runpair.sh

This will process the data and copy the result from HDFS FS to outputpair directory.

Result.txt | part-r-00000 | part-r-00001

```
29, 12)        0.31
29, 18)        0.08
29, 34)        0.31
29, 56)        0.15
29, 79)        0.08
29, 92)        0.08
34, 12)        0.25
34, 18)        0.08
34, 29)        0.25
34, 56)        0.25
34, 79)        0.08
34, 92)        0.08
```

pair's output result from reducer2

4.3 Stripe

Go to unzipped folder (hadoop) right click → Open in Terminal
type: ./runstripe.sh
This will process the data and copy the result from HDFS FS to outputstripe directory
Ex: here is the result from one of the reducer 2 output

part-r-00001

```
29      {12:0.31, 18:0.08, 34:0.31, 56:0.15, 79:0.08, 92:0.08}
34      {12:0.25, 18:0.08, 29:0.25, 56:0.25, 79:0.08, 92:0.08}
```

4.4 Hybrid

Go to unzipped folder (hadoop) right click → Open in Terminal
type: ./runhybrid.sh
This will process the data and copy the result from HDFS FS to outputhybrid directory.
output result from reducer2

part-r-00001

```
29      {12:0.31, 18:0.08, 34:0.31, 56:0.15, 79:0.08, 92:0.08}
34      {12:0.25, 18:0.08, 29:0.25, 56:0.25, 79:0.08, 92:0.08}
```

V. Comparison

	GC time elapsed (ms)	CPU time spent (ms)	Physical memory (bytes) snapshot	Virtual memory (bytes) snapshot
Pair	693	3800	696680448	7528714240
Stripe	735	4380	710283264	7528988672
Hybrid	638	3890	706478080	7529783296

Details output result can be found in Result.txt