**Projects in Hadoop**

**Add Value to Existing Data**

Concepts MapReduce, ShuffleSort, NLP Entity Extraction, Clustering

**Motivation**

How a small business can use existing data (access log, error logs, comments,

database content etc) and leverage big data technology to harvest value from it.

**Problem Scenario**

Small Business has access to a large corporate financial services database.

It wants to create a marketing campaign based on the database content but

needs to ensure the integrity of the data in the terms of uniqueness of Entity

names.

**Source Code**

**https://github.com/nsavageJVM/cluster-mapred**

**DataSets**

We have two data files

1. ComercialBanks.csv, small file for testing and development
2. ComercialBanks10k.csv larger file of 10000 lines

**Example data**

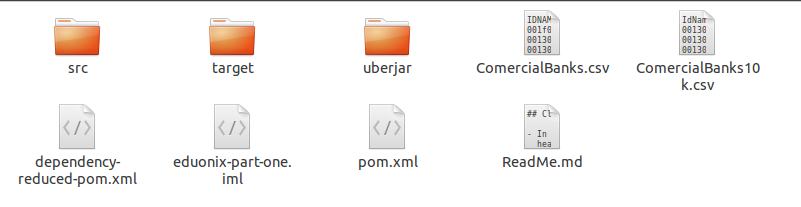
0013000000Eq2nIAAR Davenport & Company LLC 901 E Cary St FL 11 Richmond VA 23219-3745 United States

001300000063ZrhAAE **"Calamos Asset Management, Inc."** 2020 Calamos CT Ofc Naperville IL 60563-3284 United States

001300000063ZKVAA2 "Stifel Nicolaus & Company,Inc." One Financial Plaza 501 North Broadway Saint Louis MO 63102-2110 United States

001f000000Z6PqrAAF Norges Bank Bankplassen 2 Oslo 151 Norway

**Step through the code**



**Run the Code Locally from the UberJar**

**Second Video**

**Run the Code in Hadoop Cluster**

ssh root@127.0.0.1 -p 2222

hadoop

<http://hortonworks.com/hadoop-tutorial/using-commandline-manage-files-hdfs>

**// need hdfs permissions**

**su hdfs**

**// resolves to the path w.r.t. local file system**

**hadoop fs -mkdir /root**

**hadoop fs -chmod -R 777 /root**

**hadoop fs -put ComercialBanks10k.csv /root/ComercialBanks10k.csv**

**// runs in root directory local file system on vm**

**hadoop jar ubu.jar**

**hadoop fs -get /root/output\* /root**

**hadoop fs -rm -R -f /root/output\***

**Third Video**

**Step through the code**

**EntityAnalysisMRJob.java**

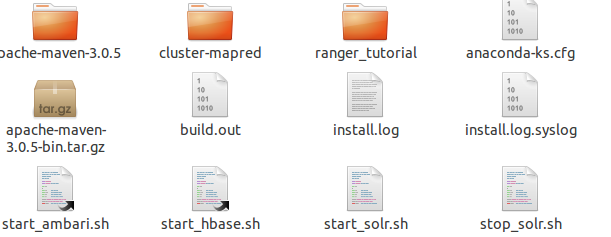
**Step through the maven build file**

**pom.xml**

**Run the code locally for testing**

**wget http://mirrors.gigenet.com/apache/maven/maven-3/3.0.5/binaries/apache-maven-3.0.5-bin.tar.gz**

**tar -zxvf apache-maven-3.0.5-bin.tar.gz**



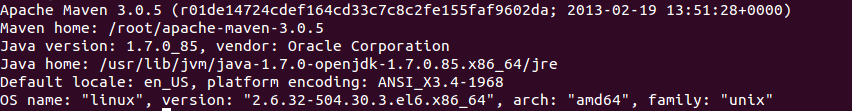
**vi .bashrc**

**export MAVEN\_HOME=/root/apache-maven-3.0.5**

**export PATH=$PATH:$MAVEN\_HOME/bin**

**source .bashrc**

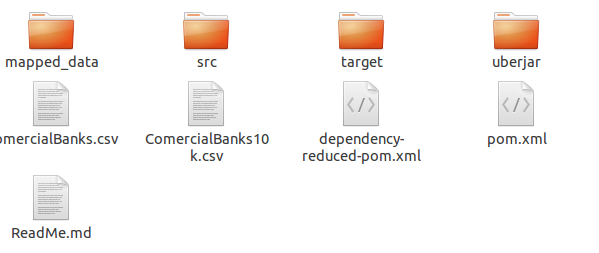
**mvn -version**



**Build the Map-Reduce Job Uber Jar**

**cd c\***

**mvn clean package -Pmapreduce**



if you have permission issues when moving and running files give read write execute to vm root directory

**cd /**

**chmod -Rf 4777 /root**

**Run the code In Hadoop**

1. change to hdfs user

**su hdfs**

1. as the hdfs user create hdfs file system and push runtime artifacts

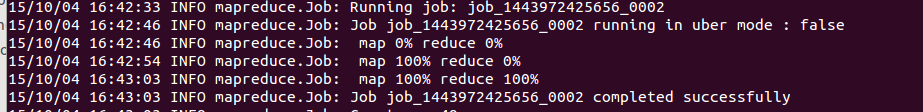
**hadoop fs -ls /**

**hadoop fs -mkdir /root**

**hadoop fs -put ComercialBanks10k.csv /root**

**hadoop fs -ls /root/**

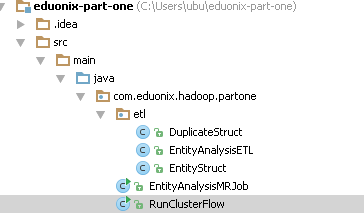
**hadoop jar ubu-mr..ar com.eduonix.hadoop.partone.EntityAnalysisMRJob**



[**http://localhost:8000/filebrowser/view/user/hue#/root/output**](http://localhost:8000/filebrowser/view/user/hue#/root/output)



**Step through the Clustering Code**

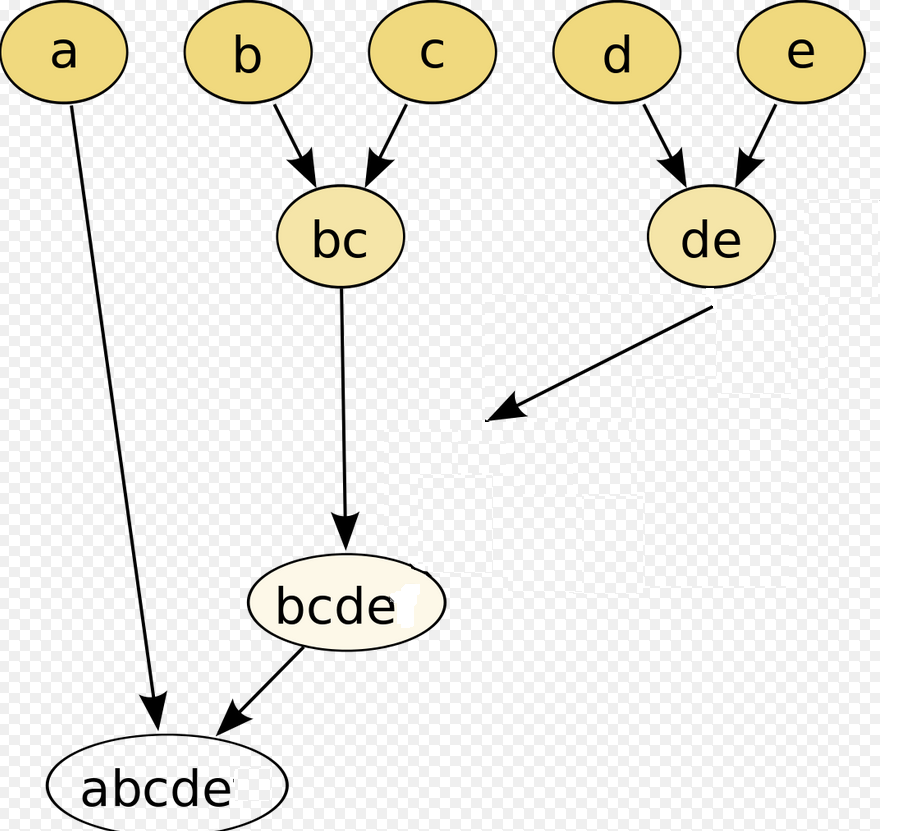


**Hierarchical Clustering**

https://en.wikipedia.org/wiki/Hierarchical\_clustering

In [data mining](https://en.wikipedia.org/wiki/Data_mining) and [statistics](https://en.wikipedia.org/wiki/Statistics), **hierarchical clustering** (also called **hierarchical cluster analysis** or **HCA**) is a method of [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) which seeks to build a [hierarchy](https://en.wikipedia.org/wiki/Hierarchy) of clusters. Strategies for hierarchical clustering generally fall into two types: [[1]](https://en.wikipedia.org/wiki/Hierarchical_clustering#cite_note-1)

* **Agglomerative**: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
* **Divisive**: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.



**Fourth Video**

**Step Through the Clustering Code**

Apache Crunch

[**https://crunch.apache.org/**](https://crunch.apache.org/)

[rockymadden](https://github.com/rockymadden)/[**stringmetric**](https://github.com/rockymadden/stringmetric)

[**https://github.com/rockymadden/stringmetric**](https://github.com/rockymadden/stringmetric)

**ETL**

A traditional ETL process extracts data cleanses, formats, and loads it for analysis. When the source data sets are large, fast, and unstructured, traditional ETL can become the bottleneck,.

**Summary Main Points**

1. Add Value to Existing Data
2. Map Reduce aggregates data in key value pairs
3. Ouput from map reduce is uniques keys aggregated values
4. Clustering aggregates data based on a metric, creates an equivalence
5. We exploit the uniques keys aggregated values as an initial clustering
6. We run a further iteration to cluster the aggregated values using a String ‘distance metric’

Clustering is a common tool for data analytics there are many ‘clustering models and algorthims’