Yatish Kelkar
TE IT 8001

Assignment 1 : Hadoop Installation

* Aim : To install Hadoop on a single node

* Theory :

Q1) What is big data? Explain various applications of big data?

Ans: i) Big data means an enormously large amount of data. This consists of large datasets that cannot be processed using the traditional computing techniques.

ii) Big data is generated from various sources, such as stock exchange data, social media data, video sharing portals, search engine data, transport data, banking data, etc

iii) It is difficult to handle big data due to various limitations of the traditional file handling systems.

iv) Big data has high levels of complexity, is produced at different velocities and has a changing level of ambiguity. Conventional processing solutions & algorithms are unable to handle such huge amounts of data.

v) Big data has various applications in the real world. They are

a) Fraud detection

- Fraud detection is a big data application for business that have operations like claims & transaction processing

- big data platforms can analyse claims & transaction processes of businesses. They identify large scale patterns or detect behaviour anomalies.

b) IT log analytics

- enormous amount of logs & of trace data is generated in IT companies. Many times this data goes unexamined.

- big data can quickly analyse & identify large scale patterns and help in diagnosing & preventing problems.

c) Call Center Analytics

- by making sense of time / quality resolution metrics, big data solutions are able to identify recurring problems of customer & staff behaviour patterns.

- big data can also process call content itself.

d) Social media analytics

- with help of social media we can gain insights into how the market is responding to product & campaigns.

- using these insights, companies can adjust their pricing, promotion & campaign placement to get optimum results.

Q2) What is Hadoop? Explain the features of Hadoop?

Ans: i) Hadoop is an open source, Java based programming framework which supports processing & storage of extremely large sets of data in a distributed computing environment using simple programming models.

ii) Hadoop has a strong processing power & ability to handle virtually unlimited number of tasks.

iii) Using Hadoop, applications can be run on systems with thousands of commodity hardware nodes. It can handle thousands of terabytes of data.

iv) Hadoop has a distributed file system (HDFS) which facilitates rapid data transfer among nodes. This allows the system to proceed even if one of the nodes fail. This avoids unexpected data loss.

v) Hadoop has emerged as a foundation for big data tasks like scientific data analysis, business & sales, social media data, etc

vi) Doug Cutting & Mike Cafarella created Hadoop in 2006 to support distribution for the Nutch search engine. In 2008, Yahoo released Hadoop as an open-source project.

Q3) Explain Hadoop ecosystem in detail. (Draw appropriate diagram)

Ans: The Hadoop ecosystem consists of following 4 components:

a) HDFS: Hadoop distributed File System states that files will be broken down into blocks and stored in nodes over the distributed architechture. It provides high thoroughput access to application data.

b) YARN: Yet Another Resource Negotiator is used for job scheduling & cluster management.

c) MapReduce: YARN based system for parallel processing of a large data set using key value pair. The map task takes input data and converts into dataset which can be computed into key-value pair.

d) Hadoop Common: These Java libraries & utilities are used to start Hadoop. They provde file system & OS level abstractions.

- Hadoop Distributed File System

i) HDFS is primary storage system used by Hadoop applications. It is a distributed file system and a framework provided by

Hadoop for analysis & transformation of huge data sets which uses the MapReduce paradigm.

ii) HDFS is based on Google File System (GFS). It provides high performance access to data to Hadoop clusters. HDFS has become a key tool to manage pools of big data & supporting big data analytics.

iii) HDFS is usually deployed on commodity hardware of low cost where possibility of server failure is common. as file system is designed to be highly fault tolerant

iv) HDFS facilitates rapid data transfers between nodes and enables Hadoop systems to precede its execution even if one of its nodes fails. This decreases risk of failure

v) HDFS uses master/slave architechture. Namenode manages metadata while datanode stores actual data. Inodes are used to represent these file & directories. File content is divided into large blocks & each block is replicated at each datanode.

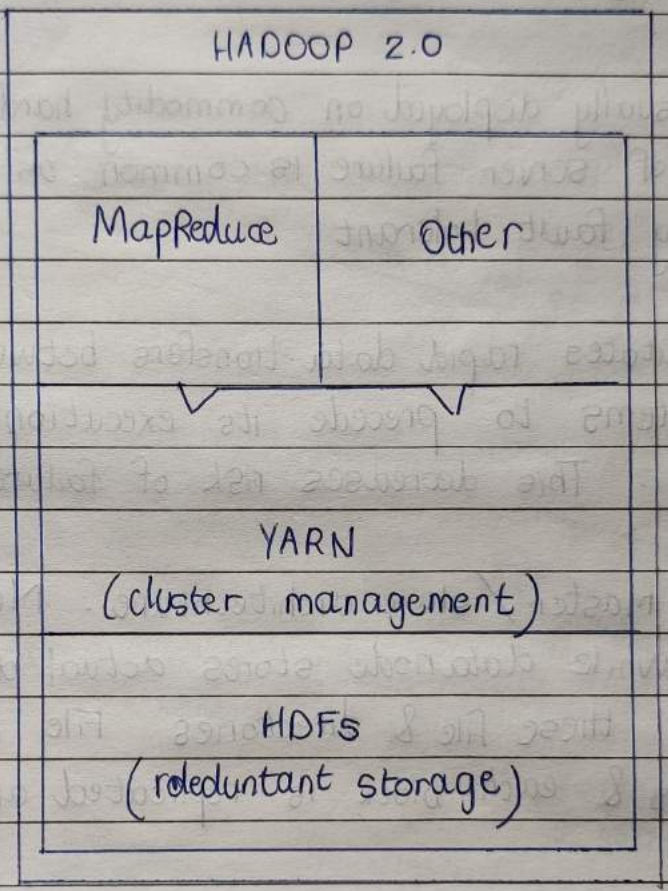
vi) Advantages of HDFS:
   a) high scalability
   b) low limitation
   c) open source
   d) low cost

vii) Disadvantages of HDFS
   a) software under active development
   b) restrictive programming model

c) high cluster management

Q4) Explain following concepts of Hadoop in detail:

1) Namen



Q4) Explain following concepts of Hadoop in detail:

1) Namenode

Ans. i) The namenode is used to manage the namespace of a system. For all files & directories, namenode preserves a file

system tree & metadata.

ii) Namenode does not hold actual data or dataset. Namenode is considered as single point of failure because if namenode is down then Hadoop cluster is not accessible to users.

iii) Namenode holds information about list of blocks & location for any given file in HDFS. Using this, namenode identifies how to build file from blocks.

iv) Namenode configuration requires a lot of RAM.

2) Datanode

Ans: i) Datanode stores actual data in HDFS. Datanode is constantly communicating with Namenode.

ii) When datanode starts it tells the namenode the list of blocks it holds.

iii) Availability of data or cluster is unaffected if datanode is down. Datanode configuration requires a lot of hard disk space.

3) Secondary Namenode

Ans: i) Secondary namenode is a dedicated node in HDFS cluster which takes checkpoints of the file system metadata available on namenode

ii) Secondary namenode stores only checkpoints & not metadata.

iii) Secondary image stores a copy of FsImage file which is used to store the snapshot of system data at particular point of time.

iv) Secondary namenode always keeps the EditLogs file small.

Q5) Explain following commands in detail (syntax & one example)

Ans: i) appendToFile: appends single src or multiple srcs from local file system to destination file system. Also reads input from stdin & appends to destination file system.

Syntax: hdfs dfs -appendToFile <localsrc> <dst>
Example: hdfs dfs -appendToFile localfile /usr/local/hadoop

ii) copyToLocal: similar to get command, except that destination is restricted to local file destination.

Syntax: hdfs dfs -copyToLocal <ignoresrc> <-crc> URI <localsrc>

iii) copyFromLocal: similar to put command, except that source is restricted to local file reference. The -f option will overwrite the destination if it already exists.

Syntax: hdfs dfs -copyFromLocal <localsrc> URI
eg: hdfs dfs -copyFromLocal /user/local/hadoop/bin URI

iv) count : counts the number of directories, files & bytes under path that matches specified pattern. Output colums are DIR_COUNT, FILE_COUNT, CONTENT_SIZE, FILE_NAME

Syntax : hdfs dfs -count [-q] <path>
Example : hdfs dfs -count -q /user/local/hadoop/bin

v) touchz : Creates a file of zero length

Syntax : hdfs dfs -touchz URI [URI ...]
Example : hdfs dfs -touchz path

vi) get : copies file to local file system. Files that fail the CRC check may be copied with the -ignorecrc flag. Files and CRCs may be copied with the -crc option.

Syntax : hdfs dfs -get [-ignorecrc][-crc] <src> <dest>
Example : hdfs dfs -get /usr/local/hadoop usr/Downloads

vii) put : copy single src or multiple srcs from local file system to destination file system. Also reads input from stdin & writes to destination file system.

Syntax : hdfs dfs -put <localsrc> .. <dst>
Example : hdfs dfs -put localfile /usr/hadoop/hadoopfile.

viii) moveFromLocal : Similar to put command, source localsrc is deleted after its copied.

Syntax: hdfs dfs - moveFromLocal <localsrc> <dst>
Example: hdfs dfs - moveFromLocal /usr/bin /usr/Downloa

ix) rm : deletes files specified as arguments. Only deletes non empty directories and files.

Syntax: hdfs dfs -rm [-skipTrash] URI [URI ..]
Example: hdfs dfs -rm hdfs://ex.com/file

x) ls : for a file returns stats on that file & for a directory returns a list of its direct children.

Syntax: hdfs dfs -ls <args>
Example: hdfs dfs -ls /usr/hadoop/file

xi) cat : copies source paths to stdout

Syntax: hdfs dfs -cat URI [URI ..]
Example: hdfs dfs -cat hdfs://ex1.com/file1
hdfs://ex3.com/file7

Q6) Write steps required for Hadoop configuration on single node.

Ans:

i) java -version

ii) sudo apt-get update

iii) sudo addgroup hadoop

iv) sudo ~~chown~~ adduser -ingroup hadoop hduser

v) sudo apt-get install openssh-server

vi) sudo adduser hduser sudo

vii) su - hduser

viii) ssh -keygen -t rsa -P " "

ix) cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

x) ssh localhost

xi) exit

xii) cd /home/student/Downloads

xiii) sudo tar -xvgzf hadoop-2.9.0.tar.gz

xiv) sudo mv hadoop-2.9.0 /user/local

xv) cd.. cd.. cd..

xvi) sudo chown -R hduser /usr/local

xvii) sudo gedit ~/.bashrc

xviii) source ~/.bashrc

xix) sudo nano /usr/local/hadoop-2.9.0/etc/hadoop/hadoop-env.sh

xx) change JAVA HOME

xxi) sudo nano /usr/local/hadoop-2.9.0/etc/hadoop/core-site.xm

xxii) sudo nano /usr/local/hadoop-2.9.0/etc/hadoop/hdfs-site.xml

xxiii) sudo nano /usr/local/hadoop-2.9.0/etc/hadoop/yarn-site.xm

xxiv) cp mapred-site.xml.template mapred-site.xml

xxv) sudo nano /usr/local/hadoop-2.9.0/etc/hadoop/mapred-site.xml

xxvi) sudo mkdir -p /usr/local/hadoop-temp

xxvii) sudo mkdir -p /hdfs/namenode

xxviii) sudo mkdir -p /hdfs/datanode

xxix) sudo chown -R hduser /usr/local/hadoop_temp

xxx) hdfs namenode -format

xxxi) start-dfs.sh

xxxii) start-yarn.sh

xxxiii) http://localhost:50070

* Conclusion: Hadoop installation ~~of~~ on single node has been completed successfully.