

CS19P16 – DATA ANALYTICS ASSIGNMENT

1. Introduction to Hadoop

Hadoop is an open source framework from Apache and is used to store process and analyze data which are very huge in volume. Hadoop is written in Java. It is used for batch / offline processing. It is being used by Facebook, Google, Yahoo, Twitter, LinkedIn, and many more. It can be scaled up just by adding nodes in the cluster. A Hadoop cluster consists of a single master and multiple slave nodes. The master node includes Job Tracker, Task Tracker, NameNode, and DataNode whereas the slave node includes DataNode and TaskTracker.

1.1 History of Hadoop

- Hadoop started with Doug Cutting and Mike Cafarella in the year 2002 when they both started to work on Apache Nutch project.
- Apache Nutch project was the process of building a search engine project that can index upto 1 billion pages.
- After a lot of research they concluded that such a system would be very expensive and that their project architecture would not be capable enough to work around with billions of pages on the web.
- So they were looking for a feasible solution which can reduce the implementation cost as well as the problem of storing and processing of large datasets.
- In 2003, Google published a paper that described the architecture of Google's distributed file system (GFS) for storing large datasets.
- This paper can solve the problem of storing very large files which were being generated because of web crawling and indexing processes.
- In 2004, Google published one more paper on the technique MapReduce, which was the solution of processing those large datasets.
- In 2005, Cutting found that Nutch is limited to only 20-to-40 node and that Nutch wouldn't achieve its potential until it ran reliably on the larger clusters.
- In 2006, Doug Cutting joined Yahoo along with Nutch project and in 2007, Yahoo successfully tested Hadoop on a 1000 node cluster and start using it.
- In 2008, Yahoo released Hadoop as an open source project to ASF(Apache Software Foundation). It successfully tested a 4000 node cluster with Hadoop.

1.2 Versions of Hadoop

- In December of 2011, Apache Software Foundation released Apache Hadoop version 1.0.
- In August 2013, version 2.06 was made available.
- Apache Hadoop version 3.0 was released in December 2017.
- In 2018, Apache Hadoop version 3.1 was released.

1.3 System Requirements

Hardware Requirements:

- Processor: Dual-core
- Memory: 8GB RAM or more for better performance.
- Storage: 50 GB of free disk space. More if handling large datasets.
- Network: High speed internet connection for downloads and updates.

Software Requirements:

- Java Development Kit (JDK): Hadoop requires java version 8 or later.
- Python: Some Hadoop tools and ecosystem components require Python.

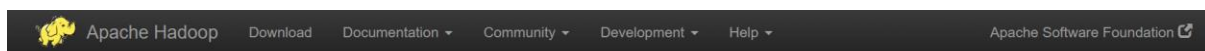
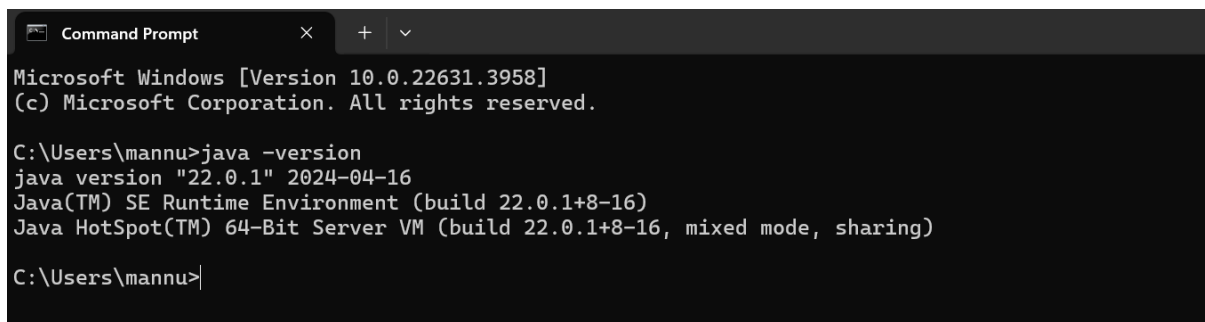
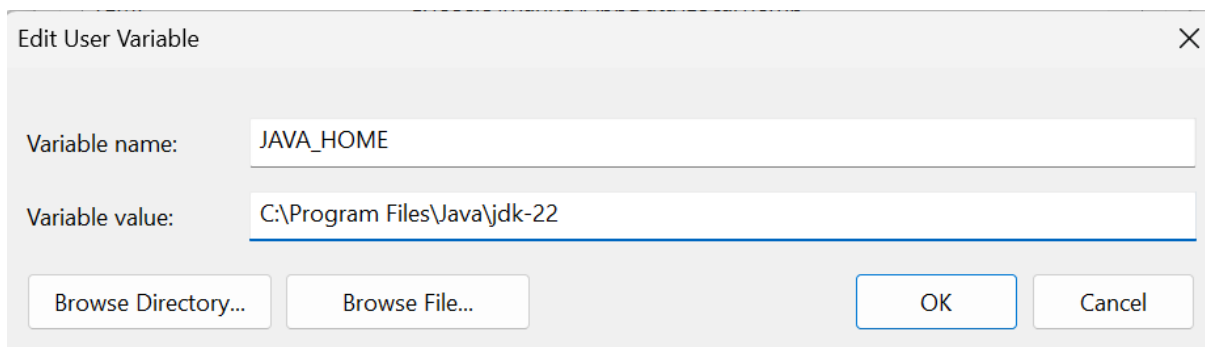
1.4 Installation and Configuration Steps

- Install java version 8 or higher.
- In the environment variables create a variable in the user variables and name it as “JAVA_HOME” and specify the JDK file path in the variable value.
- To check whether Java’s path is set properly open command prompt and type “java -version”.
- Java’s version will be displayed if the java configuration is done properly.
- Now to install Hadoop goto the official Apache Hadoop website and download the latest version.
- Extract the tar file using WinRAR into the C drive.
- After extracting the Hadoop file, open the environment variables and create a new variable “HADOOP_HOME” in the user variable with the value of the variable as the file path of the Hadoop’s bin folder
- In the Hadoop file create a new folder called “data” inside which you should create two more folders one is “namenode” and the other is “datanode” folders.
- To configure Hadoop edit the following files which are located in the etc folder :
 - core-site.xml
 - hdfs-site.xml
 - mapred-site.xml
 - yarn-site.xml
- After editing the above files open hadoop-env file in notepad and set the java home location to it.
- Now goto command prompt and type the command “hdfs namenode -format”. This command is used to format the HDFS (Hadoop Distributed File System) namenode. This command initializes the directory structure of the namenode by creating the necessary file system metadata.
- NameNode works on the Master System. The primary purpose of Namenode is to manage all the MetaData. Metadata is the list of files stored in HDFS(Hadoop Distributed File System). As we know the data is stored in the form of blocks in a Hadoop cluster. So the DataNode on which or the location at which that block of the file is stored is mentioned in MetaData. All information regarding the logs of

the transactions happening in a Hadoop cluster (when or who read/wrote the data) will be stored in MetaData. MetaData is stored in the memory.

- Now type the commands “start-dfs.cmd” and “start-yarn.cmd”. This command will start all the daemons at once.
- Daemons mean Process. Hadoop Daemons are a set of processes that run on Hadoop.
- Goto the browser and type “localhost:9870” this will open Hadoop in the browser.

1.5 Installation Screenshots



Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
3.4.0	2024 Mar 17	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.3.6	2023 Jun 23	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement

To verify Apache Hadoop® releases using GPG:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the signature file `hadoop-X.Y.Z-src.tar.gz.asc` from [Apache](#).
3. Download the [Hadoop KEYS](#) file.
4. `gpg --import KEYS`
5. `gpg --verify hadoop-X.Y.Z-src.tar.gz.asc`

Edit User Variable ✕

Variable name:

Variable value:

The screenshot shows a Windows File Explorer window with the address bar displaying the path: This PC > Windows (C:) > hadoop-3.4.0. The toolbar includes icons for file operations and a 'Sort' dropdown menu. The main area displays a list of files and folders with columns for Name, Date modified, Type, and Size. The 'data' folder is highlighted with a red rectangle.

Name	Date modified	Type	Size
bin	04-08-2024 20:20	File folder	
data	04-08-2024 15:16	File folder	
etc	04-03-2024 12:08	File folder	
include	04-03-2024 13:35	File folder	
lib	04-03-2024 13:34	File folder	
libexec	04-03-2024 13:35	File folder	
licenses-binary	04-03-2024 13:34	File folder	
logs	05-08-2024 07:59	File folder	
sbin	04-03-2024 12:08	File folder	
share	04-03-2024 14:15	File folder	
LICENSE	04-03-2024 11:14	Text Document	16 KB
LICENSE-binary	04-03-2024 11:14	File	24 KB
NOTICE	04-03-2024 11:14	Text Document	2 KB
NOTICE-binary	04-03-2024 11:14	File	27 KB
README	04-03-2024 11:14	Text Document	1 KB

<div> <div> <div></div> <div>This PC</div> </div> <div> <div></div> <div>Windows (C:)</div> </div> <div> <div></div> <div>hadoop-3.4.0</div> </div> <div> <div></div> <div>data</div> </div> </div>			
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>Sort</div> <div>View</div> <div></div> </div>			
Name	Date modified	Type	Size
<div></div> datanode	05-08-2024 07:59	File folder	
<div></div> namenode	04-08-2024 15:16	File folder	

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

core-site.xml

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.data.dir</name>
<value>file:///C:/hadoop-3.4.0/data/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:///C:/hadoop-3.4.0/data/datanode</value>
</property>
</configuration>
```

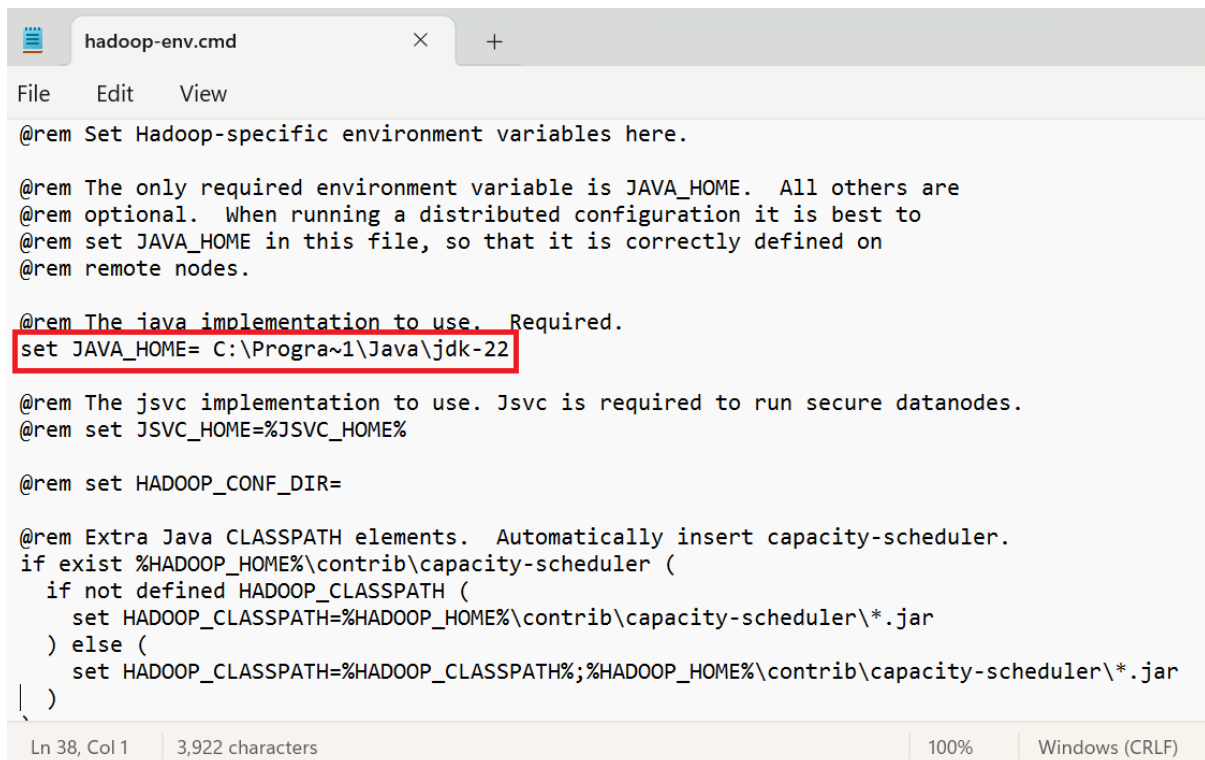
hdfs-site.xml

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

mapred-site.xml

```
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

yarn-site.xml



```
hadoop-env.cmd
File Edit View
@rem Set Hadoop-specific environment variables here.

@rem The only required environment variable is JAVA_HOME. All others are
@rem optional. When running a distributed configuration it is best to
@rem set JAVA_HOME in this file, so that it is correctly defined on
@rem remote nodes.

@rem The java implementation to use. Required.
set JAVA_HOME= C:\Progra~1\Java\jdk-22

@rem The jsvc implementation to use. Jsvc is required to run secure datanodes.
@rem set JSVC_HOME=%JSVC_HOME%

@rem set HADOOP_CONF_DIR=

@rem Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
if exist %HADOOP_HOME%\contrib\capacity-scheduler (
  if not defined HADOOP_CLASSPATH (
    set HADOOP_CLASSPATH=%HADOOP_HOME%\contrib\capacity-scheduler\*.jar
  ) else (
    set HADOOP_CLASSPATH=%HADOOP_CLASSPATH%;%HADOOP_HOME%\contrib\capacity-scheduler\*.jar
  )
)
```

Ln 38, Col 1 | 3,922 characters | 100% | Windows (CRLF)

hadoop-env file

```
Command Prompt
Microsoft Windows [Version 10.0.22631.3958]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mannu>hdfs namenode -format
2024-08-07 23:06:44,240 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = Shreeya/192.168.1.10
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 3.4.0
STARTUP_MSG: classpath = C:\hadoop-3.4.0\etc\hadoop;C:\hadoop-3.4.0\share\hadoop\common;C:\hadoop-3.4.0\share\hadoop\com
mon\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\audience-annotations-0.12.0.jar;C:\
hadoop-3.4.0\share\hadoop\common\lib\avro-1.9.2.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\bcprov-jdk15on-1.70.jar;C:\
hadoop-3.4.0\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-beanutils-1.
9.4.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-cli-1.5.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-co
dec-1.15.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop-3.4.0\share\hadoop\common\l
ib\commons-compress-1.24.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-configuration2-2.8.0.jar;C:\hadoop-3.4.0\
share\hadoop\common\lib\commons-daemon-1.0.13.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-io-2.14.0.jar;C:\hado
op-3.4.0\share\hadoop\common\lib\commons-lang3-3.12.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-logging-1.2.jar
;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-math3-3.6.1.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-net-3.9
.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\commons-text-1.10.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\curator-c
lient-5.2.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\curator-framework-5.2.0.jar;C:\hadoop-3.4.0\share\hadoop\common\
lib\curator-recipes-5.2.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\dnsjava-3.4.0.jar;C:\hadoop-3.4.0\share\hadoop\com
mon\lib\failureaccess-1.0.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\gson-2.9.0.jar;C:\hadoop-3.4.0\share\hadoop\com
mon\lib\guava-27.0-jre.jar;C:\hadoop-3.4.0\share\hadoop\common\lib\hadoop-annotations-3.4.0.jar;C:\hadoop-3.4.0\share\hado
```

```
Command Prompt
Microsoft Windows [Version 10.0.22631.3958]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mannu>start-dfs.cmd

C:\Users\mannu>start-yarn.cmd
starting yarn daemons

C:\Users\mannu>
```

Namenode information

localhost:9870/dfshealth.html#tab-overview

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Overview 'localhost:9000' (✓active)

Started:	Thu Aug 08 07:55:07 +0530 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfeec760
Compiled:	Mon Mar 04 12:05:00 +0530 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-3c7fc0a5-fa2c-4b2e-b157-b1a3214a892f
Block Pool ID:	BP-275814143-192.168.1.10-1723052211560

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 62.7 MB of 80 MB Heap Memory. Max Heap Memory is 1000 MB.

Non Heap Memory used 52.45 MB of 54.88 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	352.98 GB
Configured Remote Capacity:	0 B
DFS Used:	321 B (0%)
Non DFS Used:	154.21 GB
DFS Remaining:	198.77 GB (56.31%)
Block Pool Used:	321 B (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion (including replicas)	0

Block Deletion Start Time	Thu Aug 08 07:55:07 +0530 2024
Last Checkpoint Time	Thu Aug 08 07:53:35 +0530 2024
Last HA Transition Time	Never
Enabled Erasure Coding Policies	RS-6-3-1024k

NameNode Journal Status

Current transaction ID: 2

Journal Manager	State
FileJournalManager(root=tmp\hadoop-mannu\dfs\name)	EditLogFileOutputStream(tmp\hadoop-mannu\dfs\name\current\edits_inprogress_0000000000000000002)

NameNode Storage

Storage Directory	Type	State
tmp\hadoop-mannu\dfs\name	IMAGE_AND_EDITS	Active

DFS Storage Types

Storage Type	Configured Capacity	Capacity Used	Capacity Remaining	Block Pool Used	Nodes In Service
DISK	352.98 GB	321 B (0%)	198.77 GB (56.31%)	321 B	1

Hadoop, 2024.