

CS19P16 - DATA ANALYTICS

INTRODUCTION TO HADOOP

Introduction:

Hadoop is an open-source software framework that is used for storing and processing large amounts of data in a distributed computing environment.

It is designed to handle big data and is based on the MapReduce programming model, which allows for the parallel processing of large datasets.

Hadoop is an open-source software framework for storing a large amount of data and performing the computation.

Its framework is based on Java programming with some native code in C and shell scripts.

History of Hadoop:

Hadoop was 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 system that can index 1 billion pages. After a lot of research on Nutch, they concluded that such a system will cost around half a million dollars in hardware, and along with a monthly running cost of \$30, 000 approximately, which is very expensive. So, they realized that their project architecture will not be capable enough to the workaround 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, they came across a paper that described the architecture of Google's distributed file system, called GFS (Google File System) which was published by Google, for storing the large data sets. Now they realize that this paper can solve their problem of storing very large files which were being generated because of web crawling and indexing processes. But this paper was just the half solution to their problem.

In 2004, Google published one more paper on the technique MapReduce, which was the solution of processing those large datasets. Now this paper was another half solution for Doug Cutting and Mike Cafarella for their Nutch project. These both techniques (GFS & MapReduce) were just on white paper at Google. Google didn't implement these two techniques. Doug Cutting knew from his work on Apache Lucene (It is a free and open-source information retrieval software library, originally written in Java by Doug Cutting in 1999) that open-source is a great way to spread the technology to more people. So, together with Mike Cafarella, he started implementing Google's techniques (GFS & MapReduce) as open-source in the Apache Nutch project.

In 2005, Cutting found that Nutch is limited to only 20-to-40 node clusters. He soon realized two problems:

- (a) Nutch wouldn't achieve its potential until it ran reliably on the larger clusters
- (b) And that was looking impossible with just two people (Doug Cutting & Mike Cafarella).

The engineering task in Nutch project was much bigger than he realized. So he started to find a job with a company who is interested in investing in their efforts. And he found Yahoo!. Yahoo had a large team of engineers that was eager to work on this there project.

So in 2006, Doug Cutting joined Yahoo along with Nutch project. He wanted to provide the world with an open-source, reliable, scalable computing framework, with the help of Yahoo. So at Yahoo first, he separates the distributed computing parts from Nutch and formed a new project Hadoop (He gave name Hadoop it was the name of a yellow toy elephant which was owned by the Doug Cutting's son. and it was easy to pronounce and was the unique word.) Now he wanted to make Hadoop in such a way that it can work well on thousands of nodes. So with GFS and MapReduce, he started to work on Hadoop.

In 2007, Yahoo successfully tested Hadoop on a 1000 node cluster and start using it.

In January of 2008, Yahoo released Hadoop as an open source project to ASF(Apache Software Foundation). And in July of 2008, Apache Software Foundation successfully tested a 4000 node cluster with Hadoop.

In 2009, Hadoop was successfully tested to sort a PB (PetaByte) of data in less than 17 hours for handling billions of searches and indexing millions of web pages. And Doug Cutting left the Yahoo and joined Cloudera to fulfill the challenge of spreading Hadoop to other industries.

In December of 2011, Apache Software Foundation released Apache Hadoop version 1.0.

And later in Aug 2013, Version 2.0.6 was available.

And currently, we have Apache Hadoop version 3.0 which released in December 2017.

Versions of Hadoop:

Hadoop 1.x (Version 1)

Hadoop 2 (Version 2)

1. Hadoop 1.x

1. The Hadoop Common Module is a jar file which acts as the base API on top of which all the other components work.
2. Version one being the first one to come in existence is rock solid and has got no new updates.
3. It has a limitation on the scaling nodes with just a maximum of 4000 nodes for each cluster.
4. The functionality is limited utilizing the slot concept, i.e., the slots are capable of running a map task or a reduce task.
5. The next component if the Hadoop Distributed File System commonly known as HDFS, which plays the role of a distributed storage system that is designed to cater to large data, with a block size of 64 MegaBytes (64MB) for supporting the architecture. It is further divided into two components:

Hadoop Version 1

Name Node which is used to store metadata about the Data node, placed with the Master Node. They contain details like the details about the slave node, indexing and their respective locations along with timestamps for timelining. Data Nodes used for storage of data related to the applications in use placed in the Slave Nodes.

6. Hadoop 1 uses Map Reduce (MR) data processing model. It is not capable of supporting other non-MR tools.

MR has two components:

Job Tracker is used to assigning or reassigning task-related (in case scenario fails or shutdown) to MapReduce to an application called task tracker. The task tracker is located in the node clusters. It additionally maintains a log about the status of the task tracker. The Task Tracker is responsible for executing the functions which have been allocated by the job tracker and sends the status report of those tasks to the job tracker.

7. The network of the cluster is formed by organizing the master node and slave nodes. This cluster is further divided into racks which contain a set of commodity computers or nodes.

8. Whenever a large storage operation for a big data set is received by the Hadoop system, the data is divided into decipherable and organized blocks that are distributed into different nodes.

2. Hadoop Version 2

Version 2 for Hadoop was released to provide improvements over the issues which the users faced with version 1. Let's throw some light over the improvements that the new version provides:

HDFS Federation which has improved to provide for horizontal scalability for the name node. Moreover, the namenode was available for a single point of failure only, it is available on varied points. This is going to the Hadoop state has been increased to include the stacks such as Hive, Pig, which make this setup well equipped enabling me to handle failures pertaining to NameNode.

YARN stands for Yey Another Resource Network has been improved with the new ability to process data in the larger term that is petabyte and terabyte to make it available for the HDFS while using the applications which are not MapReduce based. These include applications like MPI and GIRAPH.

Version – 2.7.x Released on 31st May 2018: The update focused to provide for two major functionalities that are providing for your application and providing for a global resource manager, thereby improving its overall utility and versatility, increasing scalability up to 10000 nodes for each cluster.

Version 2.8.x – Released in September 2018: The updated provided improvements include the capacity scheduler which is designed to provide multi-tenancy support for processing data over Hadoop and it has been made to be accessible for window uses so that there is an increase in the rate of adoption for the software across the industry for dealing with problems related to big data.

Version 3

Below is the latest running Hadoop Updated Version

Version 3.1.x – released on 21 October 2019: This update enables Hadoop to be utilized as a platform to serve a big chunk of Data Analytics Functions and utilities to be performed over event processing alongside using real-time operations give a better result. It has now improved feature work on the container concept which enables had to perform generic which were earlier not possible with version 1.

The latest version 3.2.1 released on 22nd September 2019 addresses issues of nonfunctionality (in terms of support) of data nodes for multi-Tenancy, limitation to you only MapReduce processing and the biggest problem than needed for an alternate data storage which is needed for the real-time processing and graphical analysis.

System requirements for Hadoop:

1. Hardware Requirements

Hadoop requires sufficient hardware resources to handle large-scale data processing. Here's a breakdown of the key components and their recommended specifications:

CPU: Multi-core processors are recommended to parallelize Hadoop tasks efficiently.

RAM: At least 16GB of RAM per node is advisable. More memory ensures better performance for resource-intensive operations.

Storage: High-capacity and high-speed storage is essential. Using SSDs can significantly improve I/O operations.

Network: A high-speed network is critical for data transfer between nodes. Gigabit Ethernet or higher is recommended.

2. Software Requirements

Hadoop runs on a variety of operating systems and requires specific software configurations:

Operating System: Linux-based systems (e.g., CentOS, Ubuntu) are preferred due to better support and compatibility.

Java: Hadoop is written in Java, requiring JDK (Java Development Kit) version 8 or higher.

SSH: Passwordless SSH must be set up between the nodes for secure communication.

3. Hadoop Daemons

Hadoop consists of multiple daemons that run on various nodes in the cluster. Understanding these daemons is vital:

NameNode: Manages the metadata and namespace for HDFS.

DataNode: Stores actual data in HDFS.

ResourceManager: Allocates resources for running applications.

NodeManager: Manages resources and monitors container processes on a node.

Secondary NameNode: Handles checkpointing of the NameNode's metadata.

Installation steps:**Step 1: Download and install Java**

Hadoop is built on Java, so you must have Java installed on your PC. You can get the most recent version of Java from the official website. After downloading, follow the installation wizard to install Java on your system.

JDK: <https://www.oracle.com/java/technologies/javase-downloads.html>

Step 2: Download Hadoop

Hadoop can be downloaded from the Apache Hadoop website. Make sure to have the latest stable release of Hadoop. Once downloaded, extract the contents to a convenient location.

Hadoop: <https://hadoop.apache.org/releases.html>

Step 3: Set Environment Variables

You must configure environment variables after downloading and unpacking Hadoop. Launch the Start menu, type “Edit the system environment variables,” and select the result. This will launch the System Properties dialogue box. Click on “Environment Variables” button to open.

Click “New” under System Variables to add a new variable. Enter the variable name “HADOOP_HOME” and the path to the Hadoop folder as the variable value. Then press “OK.”

Then, under System Variables, locate the “Path” variable and click “Edit.” Click “New” in the Edit Environment Variable window and enter “%HADOOP_HOME%bin” as the variable value. To close all the windows, use the “OK” button.

Step 4: Setup Hadoop

You must configure Hadoop in this phase by modifying several configuration files. Navigate to the “etc/hadoop” folder in the Hadoop folder. You must make changes to three files: core-site.xml

hdfs-site.xml mapred-site.xml

Open each file in a text editor and edit the following properties:

In core-site.xml

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

In hdfs-site.xml

```
<configuration>  
  <property>  
    <name>dfs.replication</name>  
    <value>1</value>  
  </property>  
  <property>  
    <name>dfs.namenode.name.dir</name>  
    <value>file:/hadoop-3.3.1/data/namenode</value>  
  </property>  
  <property>  
    <name>dfs.datanode.data.dir</name>  
    <value>file:/hadoop-3.3.1/data/datanode</value> </property>
```



```
</configuration>
```

In mapred-site.xml

```
<configuration>
```

```
  <property>
```

```
    <name>mapred.job.tracker</name>
```

```
    <value>localhost:54311</value>
```

```
  </property>
```

```
</configuration>
```

Save the changes in each file.

Step 5: Format Hadoop NameNode

You must format the NameNode before you can start Hadoop. Navigate to the Hadoop bin folder using a command prompt. Execute this command:

```
hdfs namenode -format
```

Step 6: Start Hadoop

To start Hadoop, open a command prompt and navigate to the Hadoop bin folder. Run the following command:

```
start-dfs.cmd start-yarn.cmd
```

This command will start all the required Hadoop services, including the NameNode, DataNode, and JobTracker. Wait for a few minutes until all the services are started.

Step 7: Verify Hadoop Installation

To ensure that Hadoop is properly installed, open a web browser and go to <http://localhost:9870>. This will launch the web interface for the Hadoop NameNode. You should see a page with Hadoop cluster information.

Installation Screenshots:

Step 1: Download and install Java

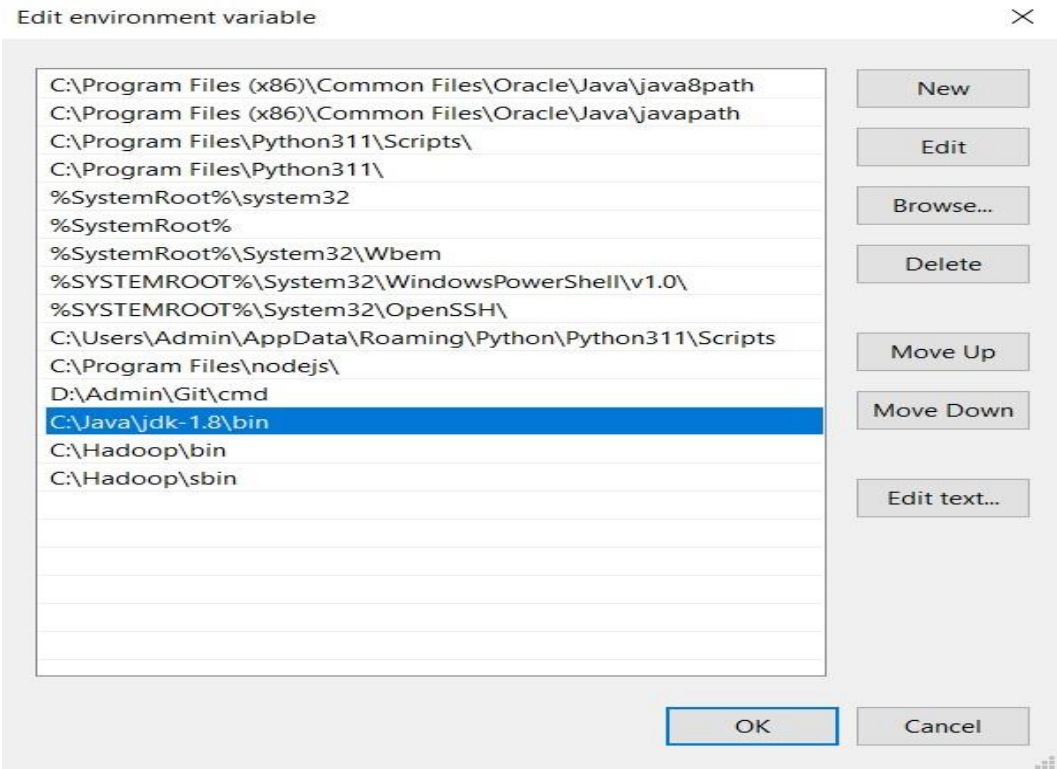
The screenshot shows the Oracle Java SE Development Kit 8u421 download page. The page includes the Oracle logo, navigation links (Products, Industries, Resources, Customers, Partners, Developers, Company), and a search bar. Below the navigation bar, there are tabs for 'Java downloads', 'Tools and resources', and 'Java archive'. The 'Java downloads' tab is selected, showing links for 'Java 8', 'Java 8 Enterprise Performance Pack', and 'Java 11'. The main content area is titled 'Java SE Development Kit 8u421' and contains several paragraphs of text, including a note about updates until December 2030, a manual update requirement for macOS, and information about the Oracle JDK 8 license change in April 2019. At the bottom, there is a table with columns for 'Product/file description', 'File size', and 'Download'. The table lists two installers: 'x86 Installer' (141.01 MB) and 'x64 Installer' (150.83 MB). The 'x64 Installer' is highlighted, and its download link is 'jdk-8u421-windows-x64.exe'.

Product/file description	File size	Download
x86 Installer	141.01 MB	jdk-8u421-windows-i586.exe
x64 Installer	150.83 MB	jdk-8u421-windows-x64.exe

The screenshot shows the 'Edit User Variable' dialog box. The 'Variable name' field is set to 'JAVA_HOME'. The 'Variable value' field is set to 'C:\Java\jdk-1.8'. There are four buttons at the bottom: 'Browse Directory...', 'Browse File...', 'OK', and 'Cancel'. The 'OK' button is highlighted.

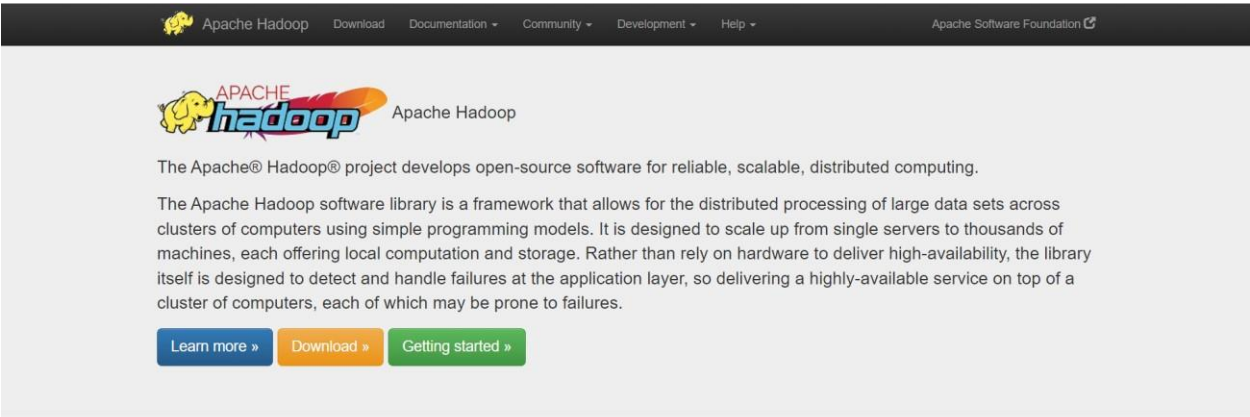
Variable name:

Variable value:



```
C:\Users\Admin>java -version
java version "1.8.0_421"
Java(TM) SE Runtime Environment (build 1.8.0_421-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.421-b09, mixed mode)
```

Step 2: Download Hadoop



Latest news

Release 3.4.0 available 2024 Mar 17

This is the first release of Apache Hadoop 3.4 line. It contains 2888 bug fixes, improvements and enhancements since 3.3.

Users are encouraged to read the [overview of main changes](#). For details of release check

Modules

The project includes these modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.

Related projects

Other Hadoop-related projects at Apache include:

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as hostnames and ability to view MapReduce, Pig and

Step 3: Set Environment Variables

Edit User Variable ✕

Variable name:

Variable value:

Edit environment variable ✕

C:\Program Files (x86)\Common Files\Oracle\Java\java8path	<input type="button" value="New"/>
C:\Program Files (x86)\Common Files\Oracle\Java\javapath	<input type="button" value="Edit"/>
C:\Program Files\Python311\Scripts\	<input type="button" value="Browse..."/>
C:\Program Files\Python311\	<input type="button" value="Delete"/>
%SystemRoot%\system32	
%SystemRoot%	
%SystemRoot%\System32\Wbem	
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\	
%SYSTEMROOT%\System32\OpenSSH\	
C:\Users\Admin\AppData\Roaming\Python\Python311\Scripts	<input type="button" value="Move Up"/>
C:\Program Files\nodejs\	<input type="button" value="Move Down"/>
D:\Admin\Git\cmd	
C:\Java\jdk-1.8\bin	
C:\Hadoop\bin	<input type="button" value="Edit text..."/>
C:\Hadoop\sbin	

```
C:\Users\Admin>hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /C:/Hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
```


Step 6: Verify Hadoop Installation

HadoopOverviewDatanodesDatanode Volume FailuresSnapshotStartup ProgressUtilities

Overview 'localhost:9000' (active)

Started:	Sun Aug 18 18:45:16 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-a23ce25d-ee9d-4000-ac1f-044f436c4c8a
Block Pool ID:	BP-934656018-192.168.56.1-1723971050909

Summary

Security is off.
Safemode is off.
19 files and directories, 5 blocks (5 replicated blocks, 0 erasure coded block groups) = 24 total filesystem object(s).
Heap Memory used 74.86 MB of 193 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 61.65 MB of 63.11 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	118.63 GB
----------------------	-----------

Non Heap Memory used 52.64 MB of 53.69 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	475.5 GB
Configured Remote Capacity:	0 B
DFS Used:	177.47 MB (0.04%)
Non DFS Used:	143.36 GB
DFS Remaining:	331.96 GB (69.81%)
Block Pool Used:	177.47 MB (0.04%)
DataNodes usages% (Min/Median/Max/stdDev):	0.04% / 0.04% / 0.04% / 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	15
Number of Blocks Pending Deletion (including replicas)	0
Block Deletion Start Time	Fri Sep 13 21:19:35 +0530 2024
Last Checkpoint Time	Fri Sep 13 21:19:36 +0530 2024
Enabled Erasure Coding Policies	RS-6-3-1024k

NameNode Journal Status

Current transaction ID: 1	
Journal Manager	State
FileJournalManager(root=C:\Hadoop\data\namenode)	EditLogFileOutputStream(C:\Hadoop\data\namenode\current\edits_inprogress_00000000000000000001)

NameNode Storage

Storage Directory	Type	State
C:\Hadoop\data\namenode	IMAGE_AND_EDITS	Active

DFS Storage Types

Storage Type	Configured Capacity	Capacity Used	Capacity Remaining	Block Pool Used	Nodes In Service
DISK	118.63 GB	148 B (0%)	21.29 GB (17.95%)	148 B	1