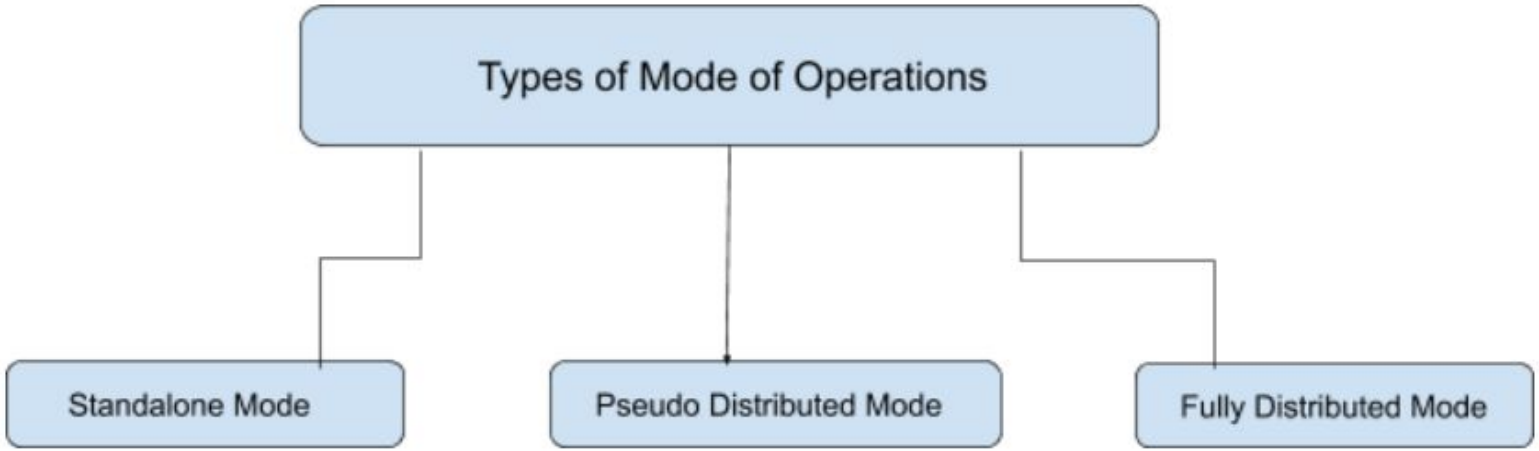
CIT650: Introduction to Big Data

Lab #1

Lab Goals

- Hadoop Installation Modes
- Hadoop on Docker
- HDFS Commands
- Hadoop UI
- Hands-on Example on MapReduce
- Lab Task

Hadoop Installation Modes

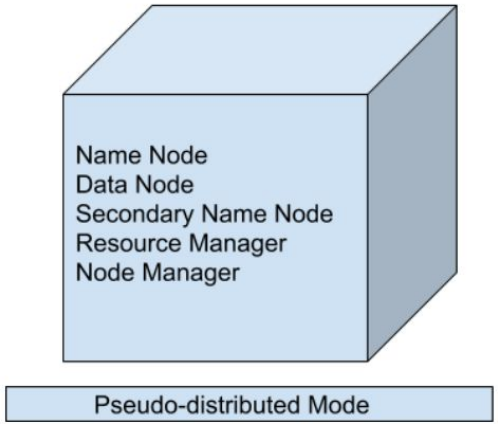


Hadoop Installation Modes: Standalone Mode

- We are installing Hadoop only in a single system. (i.e. one PC or laptop)
- We mainly use Hadoop in this Mode for the Purpose of Learning, testing, and debugging.
- In this mode, all hadoop processes will run on one JVM (Java Virtual Machine).

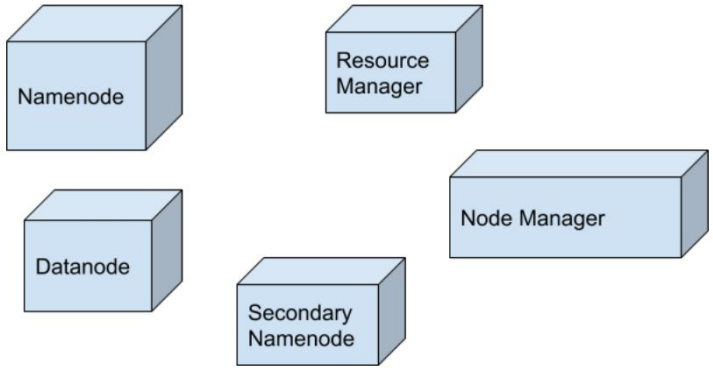
Hadoop Installation Modes: Pseudo-distributed Mode

- So called Single-node cluster.
- In this mode hadoop is to be installed also on one machine but a cluster simulated.
- All hadoop daemons will run separately on separate JVM.



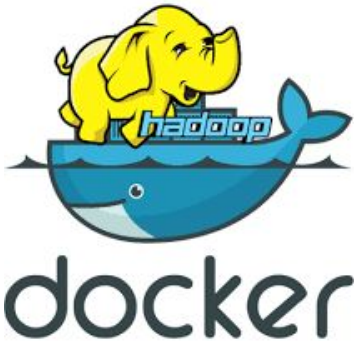
Hadoop Installation Modes: Fully-distributed Mode

- So called Multi-node cluster.
- This is the most important mode, as some nodes are used to manage the master daemons such as NameNode, and ResourceManager, other nodes for slave daemons such as DataNodes, and NodeManagers.
- Installing hadoop will be made for each node in the cluster.



Hadoop on Docker

- Due to the toughness process of hadoop installation, instead we will use Docker Container running a ready-to use hadoop in standalone mode.
- However, if you want to install it by yourself check this tutorial.
- We'll use this Docker Image: https://hub.docker.com/r/twkdocker/nuf23461-hadoop.
- Then we'll have an access to HDFS, submitting jobs, Hadoop UI, etc.



Hadoop on Docker (CONT.)

- Use the following command to install the Docker Image of Hadoop:
  - docker image pull twkdocker/nuf23461-hadoop:latest
- Then use the following command to create a Docker Container from the Image:
  - docker container run -it --name container_name twkdocker/nuf23461-hadoop:latest
- Now, you have a running container of hadoop, you have the access to the container terminal, ready for magic.

To get a list of all created containers: docker container ls -a
To start a container you have already created: docker start container_name
To get into a Docker container's shell: docker exec -it container_name bash

HDFS Commands

- To verify that hadoop and all daemons all running successfully use:
  - jps
- To list all files and directories in HDFS use:
  - hadoop fs -ls / or hdfs dfs -ls /
- To create a new directory in HDFS use:
  hadoop fs -mkdir <dir/> or hdfs dfs -mkdir <dir/>
- To copy data from local to HDFS use:
  - hadoop fs -put <local-file-path> <hdfs-file-path> or hdfs dfs -put <local-file-path> <hdfs-file-path>
- To copy data from HDFS to local use:
  - hadoop fs -get <local-file-path> <hdfs-file-path> or hdfs dfs -get <local-file-path> <hdfs-file-path>

HDFS Commands (CONT.)

- To view data of a file in HDFS use:
  - hadoop fs -cat <file-path-in-hdfs> or hdfs dfs -cat <file-path-in-hdfs>
- To move a file from one location to another in HDFS use:
  - hadoop fs -mv <source-path-in-hdfs> <dest-path-in-hdfs> or hdfs dfs -mv <source-path-in-hdfs> <dest-path-in-hdfs>
- To copy a file from one location to another in HDFS use:
  - hadoop fs -cp <source-path-in-hdfs> <dest-path-in-hdfs> or hdfs dfs -cp <source-path-in-hdfs> <dest-path-in-hdfs>
- To copy data from local to HDFS use:
  - hadoop fs -copyFromLocal <local-file-path> <hdfs-file-path> or hdfs dfs -copyFromLocal <local-file-path> <hdfs-file-path>
- To copy data from HDFS to local use:
  - hadoop fs -copyToLocal <local-file-path> <hdfs-file-path> or hdfs dfs -copyToLocal <local-file-path> <hdfs-file-path>

HDFS Commands (CONT.)

- To move data from local to HDFS use:
  - hadoop fs -moveFromLocal <local-file-path> <hdfs-file-path> or hdfs dfs -moveFromLocal <local-file-path> <hdfs-file-path>
- To move data from HDFS to local use:
  - hadoop fs -moveToLocal <local-file-path> <hdfs-file-path> or hdfs dfs -moveToLocal <local-file-path> <hdfs-file-path>
- To remove a file in HDFS use:
  - hadoop fs -rm <file-path-in-hdfs> or hdfs dfs -rm <file-path-in-hdfs>
- To create a file in a specific location in HDFS use:
  - hadoop fs -touchz <file-path-and-name-in-hdfs> or hdfs dfs -touchz <file-path-and-name-in-hdfs>

NOTE: Almost all normal terminal commands are used in HDFS.

Hadoop UI

- NameNode: localhost:9870
- DataNodes: localhost:9864/9865
- YARN: localhost:8088
- Check for more

Hands-on Example on MapReduce

- We’re going to implement a trivial example on Hadoop utilizing MapReduce for Word Count.
- Assuming we have a cluster of 100 nodes, totaling 100 TB disk space, 10 TB RAM, etc.
- We will read a text file, and we want to produce the number of occurrences of each word.
- Attached with lab slides is the Java code used and how to run the example in addition to the data file used.
- Let’s open IntelliJ.

Hands-on Example on MapReduce (CONT.)

- MapReduce jobs consists mainly of 3 classes:
  - **The Driver**: the main class of the job, contains the configurations of the job, submitting the job to the cluster, configuring Mapper and Reducer
  - **The Mapper**: The map method to specify the map behavior, each map deals with one split at a time.
  - **The Reducer**: The reduce method to specify the reduce behavior.
- We will explain the code in intelliJ.
- Add the following jars from hadoop/ to the libraries in intelliJ:
  - hadoop-common-3.3.1.jar
  - hadoop-mapreduce-client-common-3.3.1.jar
  - hadoop-mapreduce-client-core-3.3.1.jar
  - hadoop-client-api-3.3.1.jar
  - hadoop-client-runtime-3.3.1.jar
  - hadoop-hdfs-3.3.1.jar
- To know more about how to write a MapReduce Job.

Hands-on Example on MapReduce (CONT.)

- After writing your MapReduce job, we need to have the jar file of the job. [build]
- Then move it to the container. [Same for the data file]
  - docker cp your_file.jar container_name:/home/
- Move the data file from the container to the HDFS:
  - hdfs dfs -copyFromLocal <path-of-data-file> <hdfs-path>
- Finally run the following commands to submit the job:
  - hadoop jar jar_name.jar lab1/data.txt lab1/out

NOTE: Output directory shouldn’t be created before running the job.

Lab Task

- Write a MapReduce job to calculate the avg. temperature for each city in the attached .CSV file, first column is the city name, second is the temp. [**10 Marks**]
  - Driver: **2 Marks**
  - Mapper: **4 Marks**
  - Reducer: **4 Marks**
- Temperature should be converted from fahrenheit to celsius.
- Deadline is on Tuesday, 26 May. @ 11:59 PM.
- You’re required to submit the following [all as one .zip file named with your ID]:
  - Driver.java class which container your Driver, Mapper, and Reducer code.
  - The output file produced from the MapReduce job.

Contact me

- Email: alymohamed@nu.edu.eg

Thanks