Homework4- Hadoop

MapReduce on Pseudo-Distributed Hadoop

COVID-19 Data Analysis:

For all the tasks , we are assuming that the data ranges from 1st Jan 2020 to 8th April as mentioned in the assignment

Task (1) [5 points] Building on the simple WordCount example done in class and Hadoop tutorial, your task is to perform simple processing on provided COVID-19 dataset.

- The first task is to count the **total number of reported cases** for every country/location till April 8th, 2020 (NOTE: There data does contain case rows for Dec 2019, you'll have to filter that data)
- Name your program Covid19_1.java
- Program arguments description
  - The HDFS path to your input data file
  - [true | false] Include "World" data in the result or not. True will include total number of reported cases for "World" in the result, False will ignore the rows with location/country = world
  - The HDFS output path for your program. (NOTE: You should remove the output path after every execution of your program. Hadoop cannot start a job if output directory is already created)

Solution:

Command to execute the task:

1. Compile the code into byte java files
   javac -cp %HADOOP_CLASSPATH%
   C:\Users\AVEENA\eclipse-workspace\Assignment-4\src\Covid19_1.java -d
   D:\SEM2\TODB\assignment4\1\build\

2. Create jar file package
   jar -cvf D:\SEM2\TODB\assignment4\1\Covid19_1.jar -C
   D:\SEM2\TODB\assignment4\1\build\ .

3. Execute in hadoop
   hadoop jar D:\SEM2\TODB\assignment4\1\Covid19_1.jar Covid19_1
   /hadoop1/covid19_full_data.csv true /hadoop1/output/1/

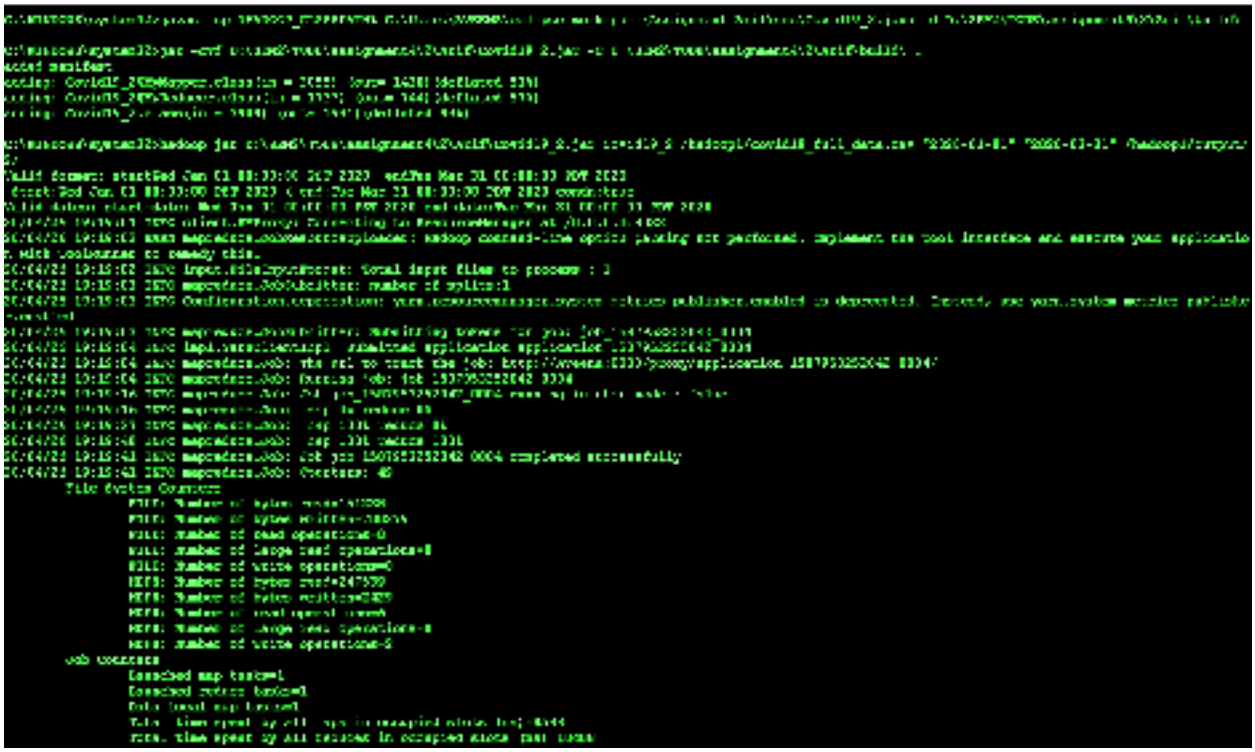
   Arguments

- First argument : The HDFS path to your input data file
- Second argument is for “world flag” . If it is true, it will include total number of reported cases for "World" in the result, else if it is False will ignore the rows with location = world
- Third argument is the HDFS output path for your program. We have program it so that it the output path exists, it will delete the existing path to avoid exceptions

(2) [10 points] General analysis usually require processing in multiple modes. For instance, for COVID-19 data, analysis can be performed on different date ranges.

- Your second task is to modify your program to report **total number of deaths** for every location/country in between a given range of dates.
  - Name your program Covid19_2.java
  - Note that you'll have to perform error handling for invalid dates. Input dataset contains data from Dec, 2019 to April, 2020. Perform your error handling accordingly
  - Also, your result should contain data including start and end dates
  - Program arguments description
    - The HDFS path to your input data
    - Start date (YYYY-MM-DD)
    - End date (YYYY-MM-DD)
    - The HDFS output path for your program. (NOTE: You should remove the output path after every execution of your program. Hadoop cannot start a job if output directory is already created)

Solution:



Command to execute the task:

1. Compile the code into byte java files
   javac -cp %HADOOP_CLASSPATH%
   C:\Users\AVEENA\eclipse-workspace\Assignment-4\src\Covid19_2.java -d
   D:\SEM2\TODB\assignment4\2\build\

2. Create jar file package
   jar -cvf D:\SEM2\TODB\assignment4\2\Covid19_2.jar -C
   D:\SEM2\TODB\assignment4\2\build\ .

3. Execute in hadoop
   hadoop jar D:\SEM2\TODB\assignment4\2\Covid19_2.jar Covid19_2
   /hadoop1/covid19_full_data.csv "2020-01-01" "2020-03-31" /hadoop1/output/2/

Arguments

- First argument : The HDFS path to your input data file
- Second argument is start date of the interval
- Third argument is end date of the interval
- Fourth argument is the HDFS output path for your program. We have program it so that it the output path exists, it will delete the existing path to avoid exceptions

Conditions for checking validity of the dates:

1. Start and end dates should contain the interval from 1st Jan 2020 to 8th April 2020. If both start and end dates are before or after, it is an invalid date interval.
2. Both the dates should have the date format - “YYY-MM-DD” . Else it is not in proper format and hence considered invalid.
3. Start date should always be before the end date

(3) [10 points] Often we can encounter cases where we have to join multiple datasets for analysis. One particular case is when one of the joining dataset is static and may have to be read frequently multiple times. In such cases, Hadoop provides DistributedCache to efficiently manage read-only files and avoid unnecessary copying.

- Your third task is to output the total number of cases per 1 million population for every country
- Name your program Covid19_3.java
- Add population.csv file to Hadoop DistributedCache (See Examples)
- Use formula
  (total_number_of_country_cases/country_population) * 1,000,000
- Program arguments description
  - The HDFS path to your input data file (covid19_full_data.csv)
  - The HDFS path to populations.csv
  - The HDFS output path for your program. (NOTE: You should remove the output path after every execution of your program. Hadoop cannot start a job if output directory is already created)
  -

Solution:

Command to execute the task:

1. Compile the code into byte java files
   javac -cp %HADOOP_CLASSPATH%
   C:\Users\AVEENA\eclipse-workspace\Assignment-4\src\Covid19_3.java -d
   D:\SEM2\TODB\assignment4\3\build\

2. Create jar file package
   jar -cvf D:\SEM2\TODB\assignment4\3\Covid19_3.jar -C
   D:\SEM2\TODB\assignment4\3\build\ .

3. Execute in hadoop
   hadoop jar D:\SEM2\TODB\assignment4\3\Covid19_3.jar Covid19_3
   /hadoop1/covid19_full_data.csv /hadoop1/populations.csv /hadoop1/output/3/

Arguments

- First argument : The HDFS path to your input data file
- Second argument is to the populations.csv file
- Third argument is the HDFS output path for your program. We have program it so that it the output path exists, it will delete the existing path to avoid exceptions

Performance Comparison:

	Hadoop	***Spark***
Task 1 : Covid19_1	4.142 seconds	2.18seconds
Task 2 : Covid19_2	3.02 seconds	1.191 seconds
Task 3 : Covid19_3	3.9 seconds	1.093 seconds

*Performance is dependent on the systems. May vary on a different OS/docker

Extra Credit:

Task 1 and task 2:

Command to execute the task:
Task 1:

spark-submit SparkCovid19_1.py /cse532/input/covid19_full_data.csv true /cse532/output/

spark-submit --master local[2] SparkCovid19_1.py /cse532/input/covid19_full_data.csv false /cse532/output/

Task 2:
spark-submit SparkCovid19_2.py /cse532/input/covid19_full_data.csv 2020-01-01 2020-03-31 /cse532/output/

Task 3:
spark-submit SparkCovid19_3.py /cse532/input/covid19_full_data.csv /cse32/input/populations.csv /cse532/output/

```
C:\>spark-submit D:\SEM2\TODB\assignment4\SPARK\SparkCovid19_2.py /hadoop1/covid19_full_data.cs
v 2020-01-01 2020-03-31 /hadoop1/output/2/
20/04/29 20:38:38 INFO spark.SparkContext: Running Spark version 2.4.5
20/04/29 20:38:38 INFO spark.SparkContext: Submitted application: SparkCovid19_2.py
20/04/29 20:38:38 INFO spark.SecurityManager: Changing view acls to: AVEENA
20/04/29 20:38:38 INFO spark.SecurityManager: Changing modify acls to: AVEENA
20/04/29 20:38:38 INFO spark.SecurityManager: Changing view acls groups to:
20/04/29 20:38:38 INFO spark.SecurityManager: Changing modify acls groups to:
20/04/29 20:38:38 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls
 disabled; users  with view permissions: Set(AVEENA); groups with view permissions: Set(); user
s  with modify permissions: Set(AVEENA); groups with modify permissions: Set()
20/04/29 20:38:39 INFO util.Utils: Successfully started service 'sparkDriver' on port 53655.
20/04/29 20:38:39 INFO spark.SparkEnv: Registering MapOutputTracker
20/04/29 20:38:39 INFO spark.SparkEnv: Registering BlockManagerMaster
20/04/29 20:38:39 INFO storage.BlockManagerMasterEndpoint: Using org.apache.spark.storage.Defau
ltTopologyMapper for getting topology information
20/04/29 20:38:39 INFO storage.BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
20/04/29 20:38:39 INFO storage.DiskBlockManager: Created local directory at C:\Users\AVEENA\App
Data\Local\Temp\blockmgr-b2aab6df-c8ea-4b8d-812e-23d97af222fd
20/04/29 20:38:39 INFO memory.MemoryStore: MemoryStore started with capacity 366.3 MB
20/04/29 20:38:39 INFO spark.SparkEnv: Registering OutputCommitCoordinator
```

```
C:\>spark-submit D:\SEM2\TODB\assignment4\SPARK\code\SparkCovid19_3.py /hadoop1/covid19_full_da
ta.csv /hadoop1/populations.csv /hadoop1/output/3
20/04/29 20:51:07 INFO spark.SparkContext: Running Spark version 2.4.5
20/04/29 20:51:07 INFO spark.SparkContext: Submitted application: SparkCovid19_3.py
20/04/29 20:51:08 INFO spark.SecurityManager: Changing view acls to: AVEENA
20/04/29 20:51:08 INFO spark.SecurityManager: Changing modify acls to: AVEENA
20/04/29 20:51:08 INFO spark.SecurityManager: Changing view acls groups to:
20/04/29 20:51:08 INFO spark.SecurityManager: Changing modify acls groups to:
20/04/29 20:51:08 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls
 disabled; users  with view permissions: Set(AVEENA); groups with view permissions: Set(); user
s  with modify permissions: Set(AVEENA); groups with modify permissions: Set()
20/04/29 20:51:09 INFO util.Utils: Successfully started service 'sparkDriver' on port 53762.
20/04/29 20:51:09 INFO spark.SparkEnv: Registering MapOutputTracker
20/04/29 20:51:09 INFO spark.SparkEnv: Registering BlockManagerMaster
20/04/29 20:51:09 INFO storage.BlockManagerMasterEndpoint: Using org.apache.spark.storage.Defau
ltTopologyMapper for getting topology information
20/04/29 20:51:09 INFO storage.BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
20/04/29 20:51:09 INFO storage.DiskBlockManager: Created local directory at C:\Users\AVEENA\App
Data\Local\Temp\blockmgr-29023d11-f0c4-4220-96ad-8b7d5c4fe195
20/04/29 20:51:09 INFO memory.MemoryStore: MemoryStore started with capacity 366.3 MB
20/04/29 20:51:09 INFO spark.SparkEnv: Registering OutputCommitCoordinator
20/04/29 20:51:09 INFO util.log: Logging initialized @4203ms
20/04/29 20:51:09 INFO server.Server: jetty-9.3.z-SNAPSHOT, build timestamp: unknown, git hash:
 unknown
20/04/29 20:51:09 INFO server.Server: Started @4325ms
20/04/29 20:51:09 INFO server.AbstractConnector: Started ServerConnector@7bc4e03e{HTTP/1.1,[htt
p/1.1]}{0.0.0.0:4040}
20/04/29 20:51:09 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
20/04/29 20:51:09 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@7cfbbb67{/
jobs,null,AVAILABLE,@Spark}
20/04/29 20:51:09 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@7e3ed842{/
jobs/json,null,AVAILABLE,@Spark}
20/04/29 20:51:09 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@61a97a93{/
jobs/job,null,AVAILABLE,@Spark}
```