Fundamentals of big data analytics

ASSIGNMENT #02 report

BS-DS(N)

**LOADING DATA**

**Connecting MongoDB with Hadoop:**

The data of size 1.01 GB was initially loaded into mongoDB using python and iterparse methods for efficiently loading of data and was downloaded and used a .json file for the implementation of **SON algorithm**. However, MongoDB was required to be connected with hadoop to generate a connection and to load data in chunks using support threshold and chunk size.

Few commands for connection of mongodb with hadoop were used.

1. First of all clone for the connected was downloaded using:

**$ git clone** [**https://github.com/mongodb/mongo-hadoop.git**](https://github.com/mongodb/mongo-hadoop.git)

1. Then installed Gradle for running jar command on hadoop in connection with mongodb
2. Change the folder from home to the mongo-hadoop folder

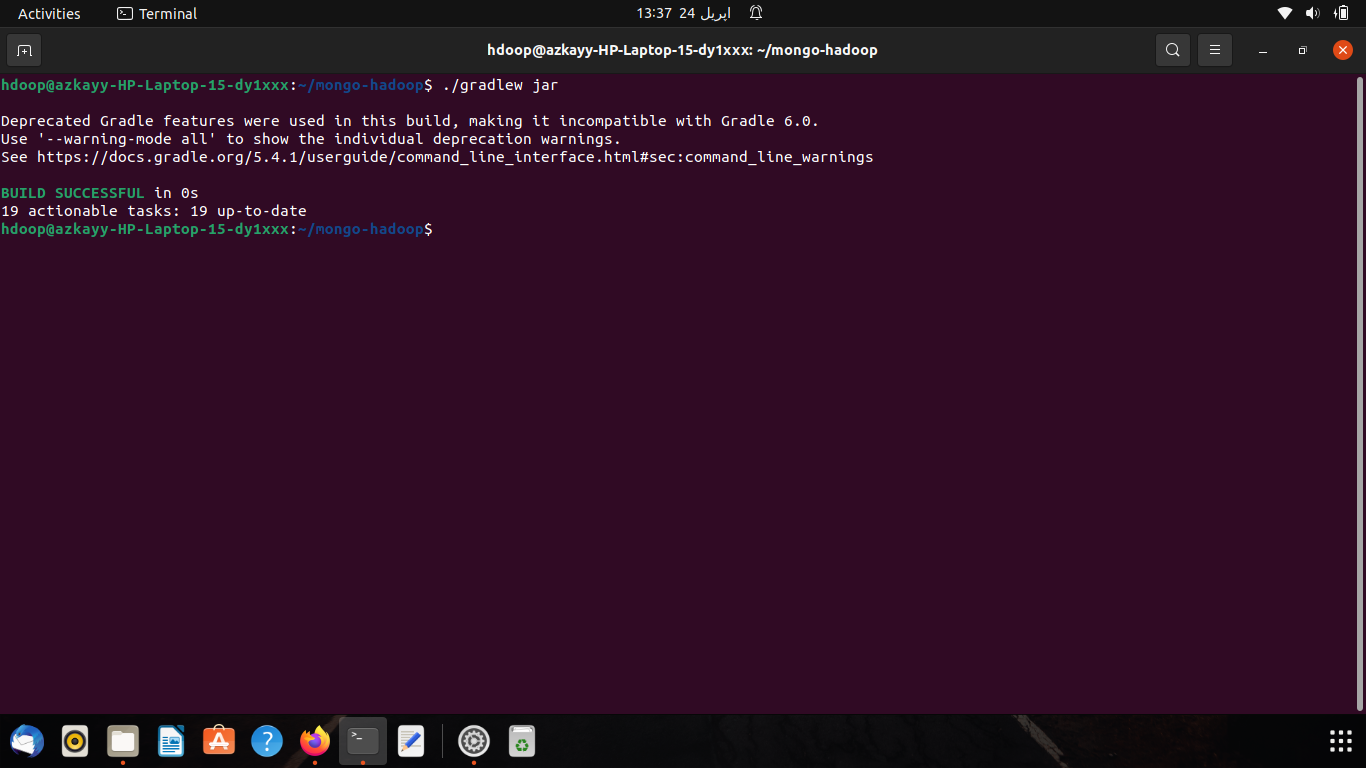
***$cd mongo-hadoop***

1. Hadoop Installation directory

***$ln –s <hadoop installation directory> ~/hadoop-binaries***

1. Run the command./ gradlew jar for the building connection through local host of mongo dB

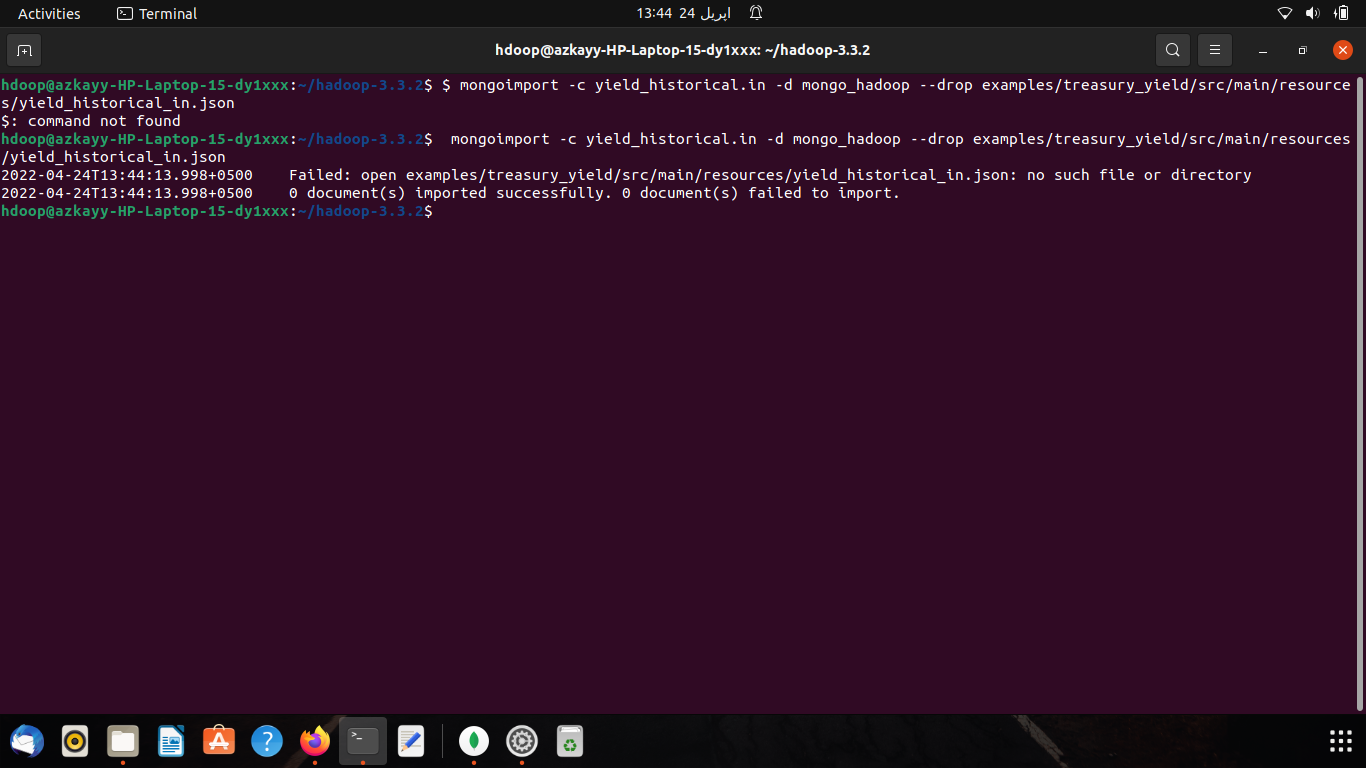
***./gradlew jar***



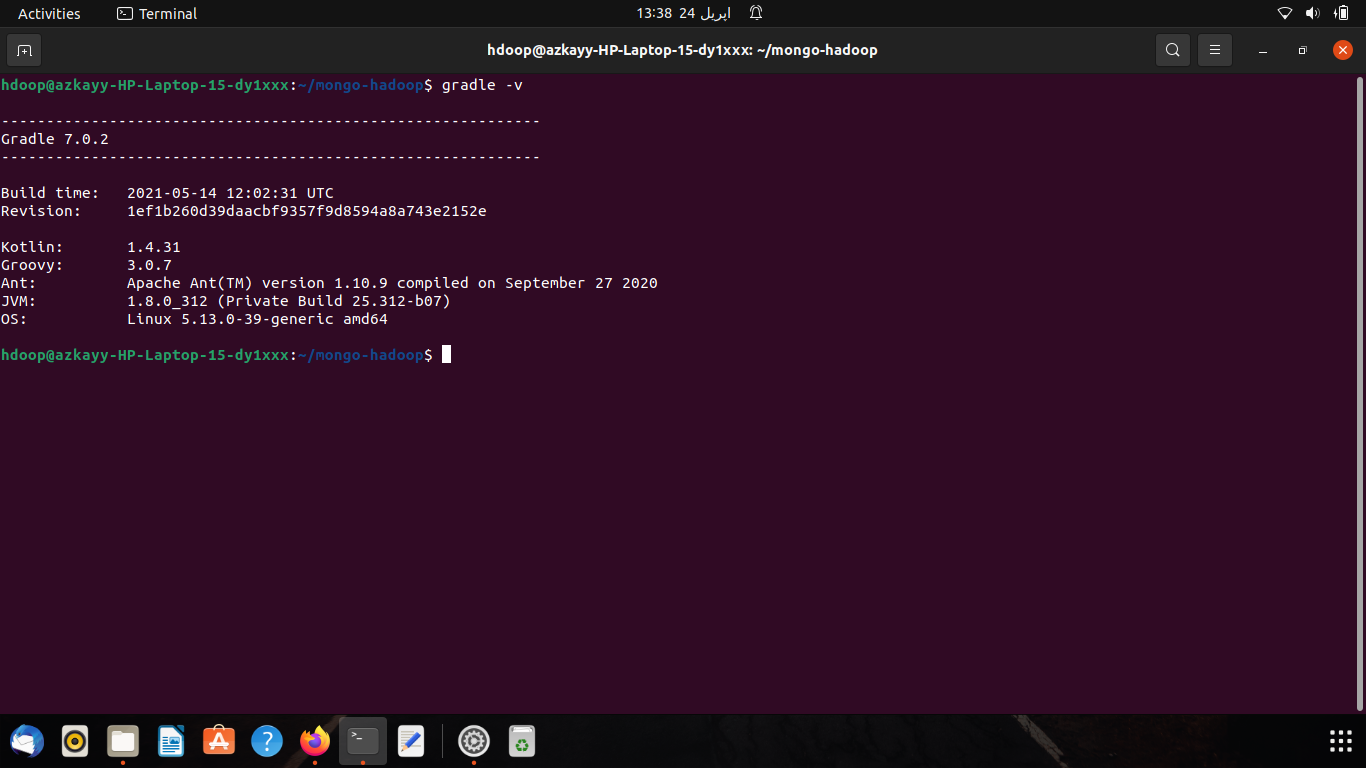
**Note: *./gradlew jar*** *commands build in portions, as while running of this command, build failed several times (probably due to connection lost). Hence this command was executed several times on terminal.*

1. Importing of MONGODB with on Hadoop server. Using:

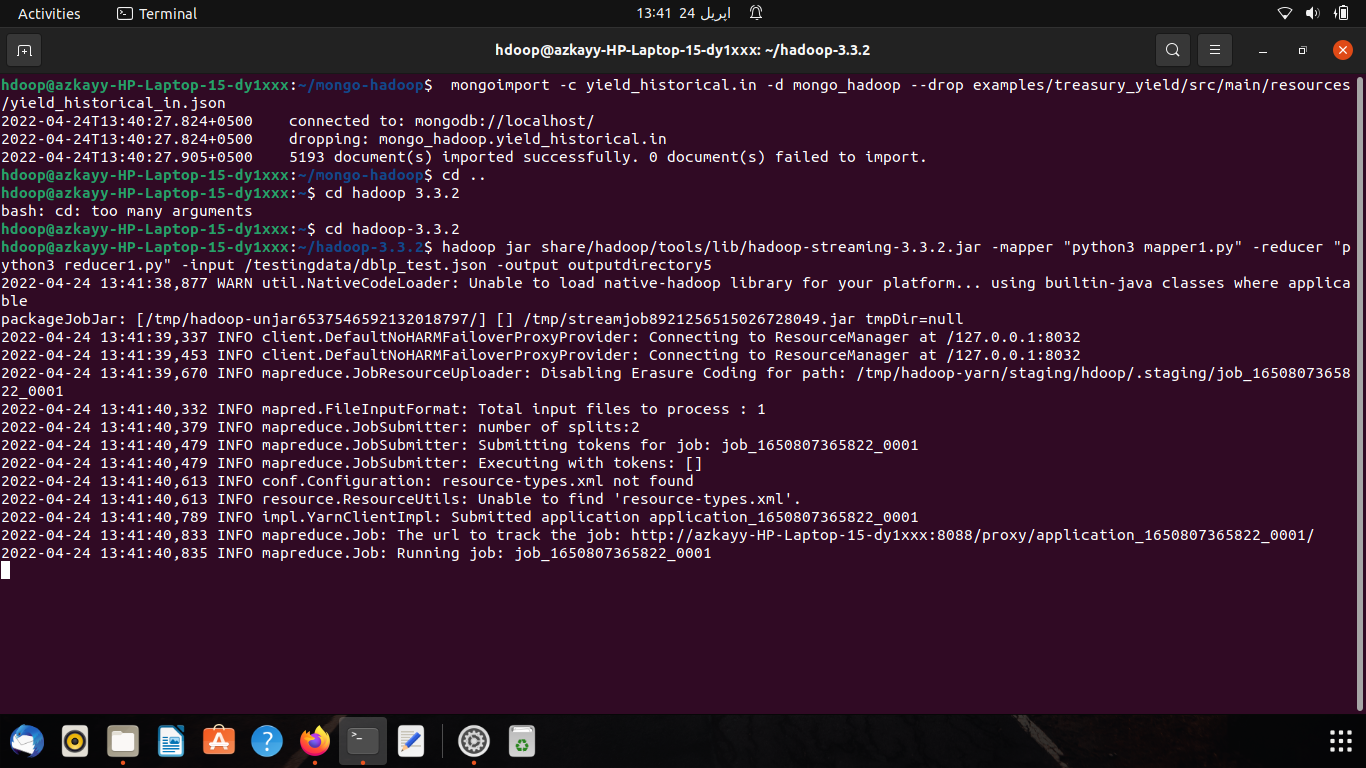
***$ mongoimport –c yield\_historical.in -d mongo\_hadoop –drop examples/treasury\_yeild/src/main/resources/yield\_historical\_in.json***



Meanwhile check the version of Gradle:

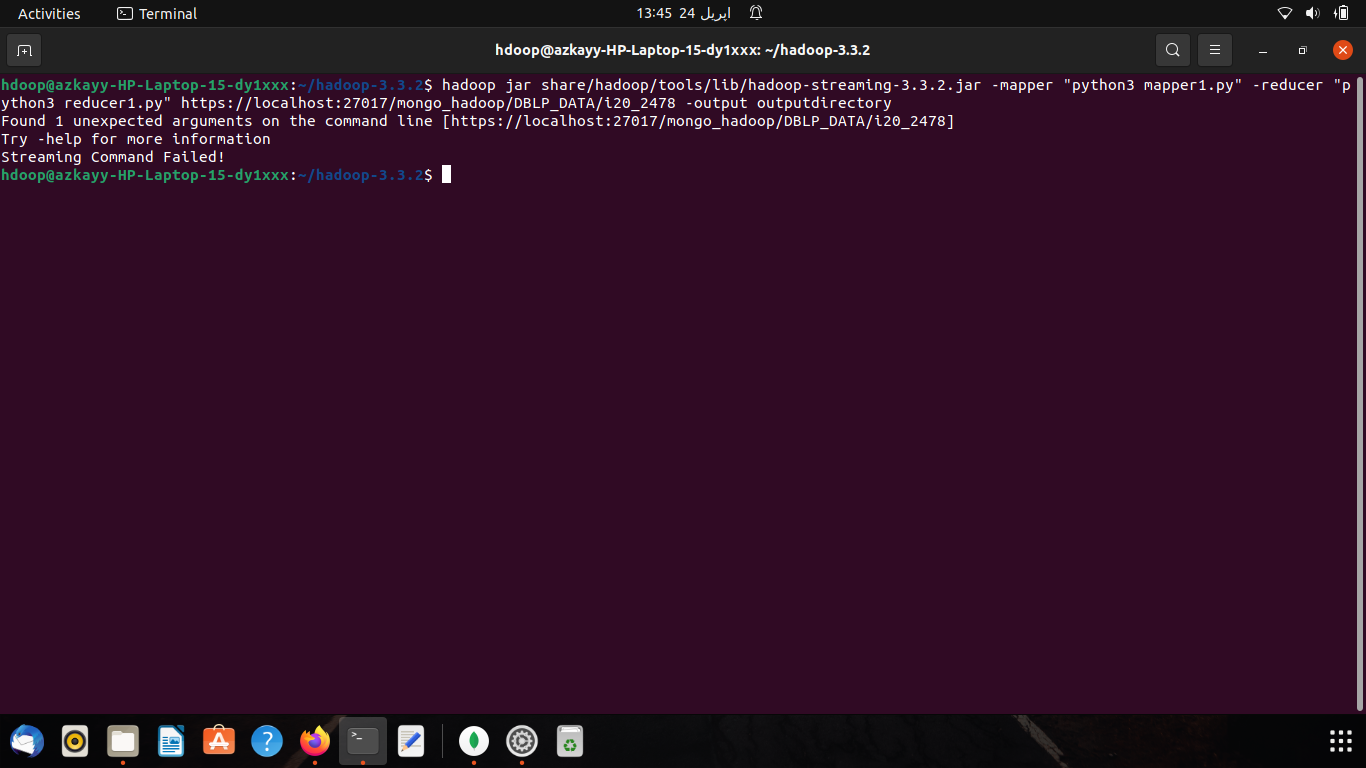


1. Running the MapReduce of DBLP\_DATA.json on Hadoop using jar command on Hadoop. As .jar folder was in the directory of hadoop-3.3.2, hence after changing directory



1. Now Starting local host of mongo dB and running the map-reduce on connection with Hadoop, However, the local host was not recognized by the Hadoop-user which in turn failed to run a map-reduce on the on connection with Hadoop.

***$Dmongo.input.uri=mongodb://localhost:27017/mongo\_hadoop.yield\_historical.inDmongo.output.uri=mongodb://localhost:27017/mongo\_hadoop.yield\_historical.out -Dmongo.input.split\_size=8 -Dmongo.job.verbose=true***



**SON ALGORITHM**

**Mapper/Reducers Explained**

**Chunk size: 8**

**Support Threshold (st): 500**

**Mapper 1:** Since the file was to be processed chunk by chunk a smaller threshold t was used, where ‘t’ was calculated to be 62.5. Considering this smaller threshold all the authors with count greater than or equals to t were considered frequent in that chunk and hence considered frequent items. Consequently, the frequent authors of all chunks were the output of mapper 1.

**REDUCER 1:** Reducer 1 uniquely identifies all the frequent authors from all the chunks, resulting in the generation of candidate items

**MAPPER 2:** Mapper 2 reads the output of reducer 1 and prints all the occurrences of each of the candidate items from the whole file

**REDUCER 2:** The job of the reducer 2 is to count the occurrences of all the authors given in the output of mapper 2 and write the names and count of those authors and their count into a file named ‘myfile1.txt’ whose count is greater than or equals to the actual support threshold i.e., 500. Those author with count less than the support threshold are left out from the file. As a result, the reducers give the names of all the frequent authors from the whole file.

**FLASK API TO DISPLAY AND FETCH THE DATA**

Application Programming Interface (API) is designed using python and flask.

Flask is one of the frameworks of the python that I have used for connecting the front end and backend. Front end is designed using html.

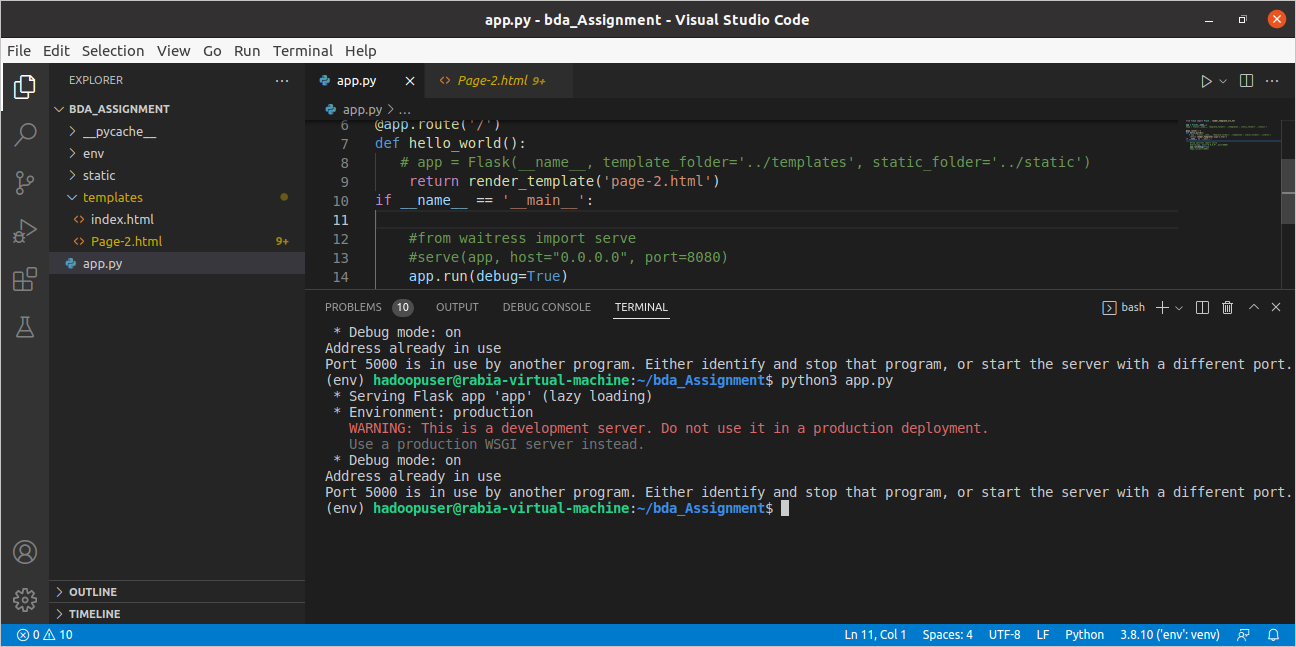
First of all, I made a html web page than I make simple connection with the flask and after that I appended my html page with the flask then after the completion of that work, I have displayed the frequent authors that was obtained from son Algorithm on the web page. (Frequent Authors were stored in a text file. Text file was red through the code and data is displayed in the form of two columns)



**ISSUSE FACED:**

One of the issues that I faced was connecting the path of the images and in appending the files.

Other than that, I started this on ubuntu made an env and then connected the flask. connection was also made correctly but after appending web page with it started giving me error (Port 5000 is in use by another program. Either identify and stop that program, or start the server with a different port.)



After surfing and trying different methods, I had resolved the issue by changing the port number but still there were some issues with port number.

other than that, it wasn't displaying web page correctly but giving Internal Server Error. after this all struggle I just shifted on windows and completed my work there.

