############### ReadMe Resubmission #################

‘It was great opportunity to learn more about Kafka and finally come up with the solution that might work as per expectation.’

Files location and conventions

All files are placed in the repository. I have enhanced consumer.py and producer.py and added more files for generator and display stats. Counting is done while consuming the records by consumer and displaying it in displayStats.py.

All the bonus questions, theory and initial setup , have been answered in previous submission in readme file (Link is below). I am not merging that here just to avoid any confusion with additional work .
https://github.com/zohaib896/Doodle (old submission)

Getting Started

I started with setting up unbuntu on AWS EC2, t2.micro server and installed all the dependencies .

The initial tasks were just related to setup and in beginning I struggled a bit with loading large file you provided and accessing it in consumer..

So there are two different ways to load file into producer and consume it ,

1. **on command terminal unbuntu as discussed previously:**

load file into producer .

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my_topic < stream.jsonl
```

read in consumer :

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092   --topic topicName --from-beginning --max-messages 100
```

I had to limit the size of messages to able to read in terminal .

2. using python

This is desirable for our counting as well and the one I worked finally. following is the explanation of each of these scripts.

Producer.py

During this step I used pandas read json function to read the file and load into producer. I created a sub data frame where I selected only the user id and time stamp .

Transformation of unix times

After this I transformed the unix time stamp into date time and added the columns of year , month , day , hours , minutes , seconds columns to the data frame . This could be used for any grouping on dataframes .

Now That I have split columns of date , I can select minutes and user id from it and select proper data structure to send it to the producer .

Selection of data structure

Firstly I thought to use dictionary of sets . Well this wasnot an option because sets are not serialize able . So I did a work around while still using sets .

I used default dictionary of lists because this can easily be send in producer.send() function. The values have to be json and serializeable for this .

However since the user ids per minute can be duplicates , i used set() data structure to make it unique and then transform it back to list and send dictionary of list to producer.send() function. Then it will contain the unique users for every minute and is serializeable .

For serialization there are other options like use of wrapper functions or libraries .

Consumer.py

Consumer parses the dictionary of unique elements . The received dictionary will have minutes as key and list of corresponding unique user ids as values .

Taking the length of each list would give the total number of users per minute . I have printed this to the stdout but I think it would be better to store this information for other stats in database .

So I have created Sqlite database and stored the data using ‘insert and replace’ statement so it will overwrite the user count (minutes is a primary key) to avoid any duplicates .

I have kept this very simple although the database abstraction can be done using sql alchemy driver and flask api .

DisplayStats.py :

So once the data is in database then I can display this information using pandas data frames using aggregates (mean , max) or group by.

This is an added feature as I have already displayed the counts in stdout .

Resolving Memory problems (partitioning vs generators ,vs pandas or Distributed framework ?) :

Initially I created small file of same data that you provided and it worked fine . However the with large files , I was facing the problem both in unbutu and pandas .

In Unbuntu its time outs and server hanging . In pandas its memory error.

To accomplish this goal I tried several things :

I try to load the data into pandas. For this there is an option to set the chunksize and lines= True   , which according to some of the stack overflow discussions should solve the memory problems . However the files was very large and doesn’t fit into memory . Later I find , this is something might be solved with creating several files in directory and then writing to these sub files and merging later or partitioning could be an option .May be distributed framework like spark would work too but requirement was not to use big data frameworks .

Using Generators

Finally I used generator to parse the file and resolve the memory error . The generator script is already present in the base directory .

Writing to new topic

This wasn’t complicated as what I did is call second producer function with new topic from consumer and send the data in stream to new topic .

What could have been done better :

The code quality can definitely be improved . I could use sql alchemy drivers or flask for data abstraction , a more object oriented code with functions classes constructors and encapsulation could be an option. There are plenty of other things that can be improved like yearly and monthly calculations in similar or different ways etc .

Conclusion and Remarks

I had a feeling of accomplishment and learning in the end, plus I was curious to learn about it too. It will keep getting better once i start to work on it more deeply. With that said I Hope for best to both of us, please do share your feedback regarding what I could improve further. And thank you very much for everything.

Further References

[1] https://medium.com/streamthoughts/streaming-data-into-kafka-s01-e03-loading-json-file-8c4c93b89ea1

[2] https://dev.to/fhussonnois/streaming-data-into-kafka-s01-e02-loading-xml-file-529i

[3] https://github.com/streamthoughts/kafka-connect-file-pulse

[4] https://stackoverflow.com/questions/38926374/how-to-get-the-all-messages-in-a-topic-from-kafka-server

[5] https://stackoverflow.com/questions/38457706/error-error-when-sending-message-to-topic

[6] https://gitlab.com/gitlab-org/omnibus-gitlab/-/issues/726

[7] https://stackoverflow.com/questions/46324067/o-apache-kafka-clients-networkclient-bootstrap-broker-hostname9092-disconne

[8] https://stackoverflow.com/questions/63985827/warn-org-apache-kafka-clients-networkclient-producer-clientid-producer-1-boo

[9] https://stackoverflow.com/questions/59426520/kafka-broker-with-no-space-left-on-device

[10] https://faust.readthedocs.io/en/latest/

[11] https://docs.docker.com/registry/deploying/

[12] https://stackoverflow.com/questions/5367118/how-perform-sqlite-query-with-a-data-reader-without-locking-database

[13] https://aiven.io/blog/create-your-own-data-stream-for-kafka-with-python-and-faker

[14] https://stackoverflow.com/questions/46524930/how-to-process-a-kafka-kstream-and-write-to-database-directly-instead-of-sending

[15] https://stackoverflow.com/questions/13861594/valueerror-invalid-literal-for-int-with-base-10

[16] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_sql.html