Google Discussions

EXAM PROFESSIONAL DATA ENGINEER TOPIC 1 QUESTION 88 DISCUSSION

Actual exam question from Google's Professional Data Engineer
Question #: 88
Topic #: 1
[All Professional Data Engineer Questions]

Your team is responsible for developing and maintaining ETLs in your company. One of your Dataflow jobs is failing because of some errors in the input data, and you need to improve reliability of the pipeline (incl. being able to reprocess all failing data). What should you do?

A. Add a filtering step to skip these types of errors in the future, extract erroneous rows from logs.

B. Add a try…€¦ catch block to your DoFn that transforms the data, extract erroneous rows from logs.

C. Add a try…€¦ catch block to your DoFn that transforms the data, write erroneous rows to Pub/Sub PubSub directly from the DoFn.

D. Add a try…€¦ catch block to your DoFn that transforms the data, use a sideOutput to create a PCollection that can be stored to Pub/Sub later.

Show Suggested Answer

by AWSandeep at *Sept. 3, 2022, 1:58 p.m.*

Comments

Type your comment...

Submit

midgoo Highly Voted 1 year, 8 months ago

Selected Answer: D

C is a big NO. Writing to PubSub in DoFn will cause bottleneck in the pipeline. For IO, we should always use those IO lib (e.g PubsubIO)
Using sideOutput is the correct answer here. There is a Qwiklab about this. It is recommended to do that lab to understand more.

upvoted 13 times

jonathanthezombieboy Highly Voted 1 year, 8 months ago

Selected Answer: D

Based on the given scenario, option D would be the best approach to improve the reliability of the pipeline.

Adding a try-catch block to the DoFn that transforms the data would allow you to catch and handle errors within the pipeline. However, storing erroneous rows in Pub/Sub directly from the DoFn (Option C) could potentially create a bottleneck in the pipeline, as it adds additional I/O operations to the data processing.

Option A of filtering the erroneous data would not allow the pipeline to reprocess the failing data, which could result in data loss.

Option D of using a sideOutput to create a PCollection of erroneous data would allow for reprocessing of the failed data and would not create a bottleneck in the pipeline. Storing the erroneous data in a separate PCollection would also make it easier to debug and analyze the failed data.

Therefore, adding a try-catch block to the DoFn that transforms the data and using a sideOutput to create a PCollection of erroneous data that can be stored to Pub/Sub later would be the best approach to improve the reliability of the pipeline.

upvoted 8 times

Farah_007 Most Recent 6 months, 4 weeks ago

Selected Answer: D

I think it's D because here you can write data from Dataflow PCollection to pub/sub.
https://cloud.google.com/dataflow/docs/guides/write-to-pubsub

upvoted 2 times

Mathew106 1 year, 3 months ago

Selected Answer: C

Answer is C. Here is the github repo and an example from the Qwiklab where they tag the output as 'parsed_rows' and 'unparsed_rows' before they send the data to GCS. I don't see how GCS or PubSub would make a difference at this point. It seems like a more maintanable solution to just parse the data in the DoFn.

1) If the function does more than that then it serves multiple purposes and it's not good software engineering. Unless there is a good reason, writing to PubSub should be separated from the DoFn.

ii) It's faster to write in mini-batches or one batch than stream the errors. What's the need for streaming out errors 1 by 1? Literally no real advantage.

https://github.com/GoogleCloudPlatform/training-data-analyst/blob/master/quests/dataflow_python/7_Advanced_Streaming_Analytics/solution/streaming_minute_traffic_pipeline.py

upvoted 1 times

tibuenoc 1 year, 7 months ago

Selected Answer: D

Output errors to new PCollection – Send to collector for later analysis (Pub/Sub is a good target)

upvoted 2 times

musumusu 1 year, 8 months ago

Option D is right approach to use to get errors as sideOutput. Apache beam has a special scripting docs not dynamic as python itself. So lets follow standard sideOutput(withoutputs in the code)
syntax be like in pipeline:
'ProcessData' >> beam.ParDo(DoFn).with_outputs

upvoted 3 times

musumusu 1 year, 8 months ago

After using you try: Catch: you can also send the erroneous records to dead letter sink into BQ
```` outputTuple.get(deadLetterTag).apply(BigQuery.write(...)) ```

upvoted 1 times

abwey 1 year, 9 months ago

Selected Answer: D

blahblahblahblahblahblahblahblah
````

upvoted 3 times

waiebdi 1 year, 9 months ago

Selected Answer: D

It`s D.
Use a try catch block to direct erroneous rows into a side output. The PCollection of the side output can be sent efficiently to the PubSub topic via Apache Beam PubSubIO.

It's not C because C means to sent every single invalid row in a separate request to PubSub which is very inefficient when working with Dataflow as now batching is involved.

upvoted 2 times

zellck 1 year, 11 months ago

Selected Answer: C

C is the answer.

upvoted 1 times

hauhau 1 year, 11 months ago

C
D: dataflow to pub/sub is weird

upvoted 1 times

Atnafu 1 year, 11 months ago

D
Side output is a great manner to branch the processing. Let's take the example of an input data source that contains both valid and invalid values. Valid values must be written in place #1 and the invalid ones in place#2. A naive solution suggests to use a filter and write 2 distinct processing pipelines. However this approach has one main drawback - the input dataset is read twice. If for the mentioned problem we use side outputs, we can still have 1 ParDo transform that internally dispatches valid and invalid values to appropriate places (#1 or #2, depending on value's validity).

https://www.waitingforcode.com/apache-beam/side-output-apache-beam/read#:~:text=simple%20test%20cases.-,Side%20output%20defined,-%C2%B6

upvoted 3 times

sfsdeniso 1 year, 11 months ago

Answer is D

upvoted 1 times

cloudmon 1 year, 12 months ago

Selected Answer: C

It's C.
In D, "storing to PubSub later" doesn't really make sense.

upvoted 2 times

devaid 2 years ago

Selected Answer: C

Answer is C. You need to reprocess all the failling data, and yes, you can use PubSub as a sink, according to the documentation: https://beam.apache.org/documentation/io/connectors/

upvoted 2 times

nickyshil 2 years, 1 month ago

Answer C

upvoted 4 times

nickyshil 2 years, 1 month ago

The error records are directly written to PubSub from the DoFn (it's equivalent in python).
You cannot directly write a PCollection to PubSub. You have to extract each record and write one at a time. Why do the additional work and why not write it using PubSubIO in the DoFn itself?
You can write the whole PCollection to Bigquery though, as explained in

Reference:
https://medium.com/google-cloud/dead-letter-queues-simple-implementation-strategy-for-cloud-pub-sub-80adf4a4a800

upvoted 6 times

AWSandeep 2 years, 2 months ago

Selected Answer: D

D. Add a try-catch block to your DoFn that transforms the data, use a sideOutput to create a PCollection that can be stored to Pub/Sub later.

upvoted 3 times