Google Discussions

EXAM PROFESSIONAL DATA ENGINEER TOPIC 1 QUESTION 277 DISCUSSION

Actual exam question from Google's Professional Data Engineer
Question #: 277
Topic #: 1
[All Professional Data Engineer Questions]

You are designing a real-time system for a ride hailing app that identifies areas with high demand for rides to effectively reroute available drivers to meet the demand. The system ingests data from multiple sources to Pub/Sub, processes the data, and stores the results for visualization and analysis in real-time dashboards. The data sources include driver location updates every 5 seconds and app-based booking events from riders. The data processing involves real-time aggregation of supply and demand data for the last 30 seconds, every 2 seconds, and storing the results in a low-latency system for visualization. What should you do?

A. Group the data by using a tumbling window in a Dataflow pipeline, and write the aggregated data to Memorystore.

B. Group the data by using a hopping window in a Dataflow pipeline, and write the aggregated data to Memorystore.

C. Group the data by using a session window in a Dataflow pipeline, and write the aggregated data to BigQuery.

D. Group the data by using a hopping window in a Dataflow pipeline, and write the aggregated data to BigQuery.

Show Suggested Answer

by scaenruy at *Jan. 4, 2024, 5:14 a.m.*

Comments

Type your comment...

Submit

raaad Highly Voted 1 year, 3 months ago

Selected Answer: B

- Hopping Window: Hopping windows are fixed-sized, overlapping intervals.
- Aggregate data over the last 30 seconds, every 2 seconds, as hopping windows allow for overlapping data analysis.
- Memorystore: Ideal for low-latency access required for real-time visualization and analysis.

upvoted 12 times

anushree09 1 year ago

Hopping windows are sliding windows. It makes sense to use that over tumbling (fixed) window because the ask is to collect last 30 seconds of data every 5 second

upvoted 2 times

Jeyaraj Most Recent 9 months, 2 weeks ago

OPTION A. (IGNORE MY Previous Comment)

Tumbling windows are the best choice for this ride-hailing app because they provide accurate 2-second aggregations without the complexities of overlapping data. This is crucial for real-time decision-making and ensuring accurate visualization of supply and demand.
Hopping windows introduce potential inaccuracies and complexity, making them less suitable for this scenario. While they can be useful in other situations, they are not the optimal choice for real-time aggregation with strict accuracy requirements.

upvoted 1 times

Jeyaraj 9 months, 2 weeks ago

Option B.

Tumbling windows are the best choice for this ride-hailing app because they provide accurate 2-second aggregations without the complexities of overlapping data. This is crucial for real-time decision-making and ensuring accurate visualization of supply and demand.
Hopping windows introduce potential inaccuracies and complexity, making them less suitable for this scenario. While they can be useful in other situations, they are not the optimal choice for real-time aggregation with strict accuracy requirements.

upvoted 1 times

JyoGCP 1 year, 2 months ago

Selected Answer: B

Option B

upvoted 1 times

ashdam 1 year, 2 months ago

hopping window is clear but memorystore vs bigquery?? Why memorystore and not bigquery?

upvoted 1 times

ML6 1 year, 2 months ago

Memory store is an in-memory key-value database for use cases such as real-time application.

upvoted 1 times

ea2023 1 year ago

Let me complete your answer MS vs BQ in this case is a matter of low latence where MS is the winner but if precision were stated about a large amount of data BQ then would"ve been the best choice.

upvoted 1 times

Jordan18 1 year, 3 months ago

why not D?

upvoted 1 times

RenePetersen 1 year, 2 months ago

Because BigQuery is not a low latency system...

upvoted 1 times

scaenruy 1 year, 4 months ago

Selected Answer: B

B. Group the data by using a hopping window in a Dataflow pipeline, and write the aggregated data to Memorystore.

upvoted 2 times