Dataflow - retries at activity level for intermittent failures

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

Contents

Issue

Many times customer's Dataflow job fails with intermittent 'SystemErrors', we suggest customers to set activity level retry for this. Customers usually have concerns regarding this, like if a task runs twice then there could be duplicate data. This TSG is explaining more about retries.

Background

Dataflow uses spark to run the jobs. Spark runs on the clusters which are set of virtual machines. Sometimes we see failure due to intermittent issues in preparing clusters. We suggest customers to have retry at the activity level to guard their workflows against such intermittent errors. This is what customers can do from their side. From the PG side, we keep track of the failures and act on the failures based on the number and type of such failures.
Retry policy at the activity level does not distinguish between 'UserError' and 'SystemError'. So this retry policy is applicable to all kinds of failures, this also adds to the user's concern.

Understanding retry process in the context of sinks

Usually, this concern about writing the duplicate data comes in the case of SQL sinks. Few insights on how sinks work

- Data in SQL kind of sinks are written in a single transaction. Like when data is being written to the target table before that data is staged in the temporary/staging table. After data is written to temporary tables, it is moved to target tables. So in case, there is a failure while writing to sink data either there will be data or not there. So duplication won't happen in this scenario due to retry.
- When there are multiple sinks then data is committed to tables one by one. In case there is a failure after writing to the sink, and while writing to the second sink, then first sink will be repeated again, and duplicate data can be written in such scenario.

Types of failures and retries

Failures can happen due to different stages and due to different root causes.

1. System Errors: Most of the system errors we see are during the start of the job. This is the main point for which we ask users to retry the activity. In this case, job never started, so duplicate data can't happen. If a system error happens during the run, which is very rare, then we can see the job running again.
2. User Errors: User errors can be of 2 types. Permanent - like bad credentials, even on retries this job will not succeed. User errors can also be transient, like SQL server under load and not able to run the query fully and fail, these kinds of runs can end successfully in subsequent retries. So in the second kind of intermittent errors which happen during a job is running can result in duplicate data.

Does handling for retries required

Most of the time retries won't require any special handling, as percentage-wise it is rare that dataflow will fail in the middle and would end up writing duplicates in rerun. The user should check the priority/effort on his side, either do the manual operation in case of duplicate data to clean up or do manual operations in case of failures.

How users can avoid duplicates

Dataflows can be designed to avoid duplicates during the reruns. Like user can always query the target tables and if rows exist then try row updates rather than inserts. Also, the user can mark all the rows as Upserts so that rows are always being updated in case those exist. There are other scenarios as well like the user was trying to delete the rows that are already deleted in the previous run so while considering design for retries for such scenarios user has to make sure if rows are already deleted (by lookup this can be checked), ignore the incoming rows. These are just hints, the actual user environment affects these solutions.

How good have you found this content?

🙂 🙁