[Cosmos DB] Heavy workload caused HttpRequestException

Last updated by | Veena Pachauri | Mar 8, 2023 at 11:59 PM PST

Contents

Issue

```
Error message-
{"dataRead":141568240360,"dataWritten":216269987696,"filesRead":126,"sourcePeakConnections":48,"sinkPeakConnections":64,"rowsRead":373003851,"rowsCopied":3707
```

Analysis

You can leverage such kusto query to get dataRead telemetry:

CustomLogEvent | where ActivityId == "runid" and TraceMessage == "TransferServiceExecutorExecutionState"

In this case, we can see "dataRead":141568240360 more than 100+G data.

"usedDataIntegrationUnits":64,"usedParallelCopies":64

Solution

First I'd like to clarify that such transient IO/Socket exception is not a VM wide connectivity issue. It is more like a transient network reliability issue between VM and Cosmos DB server.

In ADF client side, we are using the Cosmos DB SDK to access the Cosmos DB server. For such transient IO/Socket exception, the SDK should have the retry logic to handle them well. But for now, we are not sure about it, and we need to further reach Cosmos DB team to confirm that. If there is anything miss, a newer version of SDK would be needed for better resilience.

Considering the above enhancement would take time, here are the suggestions to work around the current resilience issue:

• First is to make sure the copy activity retriable. We notice that the write behavior is insert, which means a retried run would generate dup records. Suggest to enhance the copy activity to use upsert instead to make the copy activity retriable.

```
    o With that, you can set the copy activity's retry count, and a failed run would be auto retried.
```

you can get "writeBehavior":"insert" info with kusto below:

CustomLogEvent | where ActivityId == "runid" and TraceMessage == "TransferServiceExecutorJobPayload"

• Then is how to further mitigate the high fail rate:

```
o       The current used copy setting is 64 DIUs/parallelCopies/maxConnections. The transient IO/Socket exception may be caused by too heavy workloads, and it
o       Currently each copy run handles 100+ GB data. This makes any failure would re-copy such much data. It is suggested to further partition the data set (
```

Reference

120050721000949

https://icm.ad.msft.net/imp/v3/incidents/details/188164406/home https://icm.ad.msft.net/imp/v3/incidents/details/188603299/home

How good have you found this content?