Dataflow Out of Memory (OOM) Issues

Last updated by | Anudeep Sharma | Feb 24, 2023 at 10:09 PM PST

Contents

Intention

As you know OOM issue is widespread and the reason for OOM are many. We will keep on improvising this wiki with more scenarios.

https://supportability.visualstudio.com/AzureDataFactory/_wiki/wikis/AzureDataFactory/857242/Introduction-to-different-concepts-in-Mapping-Dataflows

Please note that Error 143 is also OOM. These 143 or OOM may not occur due to Outages.

Dataflows/Spark divides the data into partitions and transforms it using different processes. If the data size in a partition is more than the process can hold in memory then the process fails with OOM. So for out of memory basically we are trying to find what is causing a lot of data in the partitions (single or more).

How to narrow down on the transformation(s) causing the issue?

OOM issues are best debugged from DataflowClusterLogs. Search logs with ActivityRunId (please don't use * as ClusterTrackingId is same as ActivityRunId and is used across multiple activities).

```
DataflowClusterLogs | where ActivityRunId == "4ab98195-3736-4b62-9f29-ee6affb7d96c"
```

Search from transformedEvent or transformingEvent in message column, look for last such message, and see which stage failed and what transformations are there. These are the transformations which you should be looking in to. If you see no transformedEvent being reported then issue is on source like SQL source etc, where a lot of data is being pulled in to single partition.

In case you see number of rows are good in last transformedEvent, then check for what is being done after that like if a merge files operations is logged, usually merge files are not supposed to give OOM, but we have seen corner cases where it happens.

Below TSG provides more insights for the transformedEvents, checkout for more information. https://supportability.visualstudio.com/AzureDataFactory/_wiki/wikis/AzureDataFactory/857325/Dataflow-Timeouts-slower-runs-long-duration-etc

Compare with other runs

Event without asking customer you can get the other runs for same activity by running following function GetDataflowActivityBeforAfterRunInformation(activityID) Above will provide more information of other runs, so that you can compare those among each other.

TIMESTAMP	pipelineName	activityName	activityRunId	status	errorCode	effectiveIntegrationRunti...	duration	BeforeOrAfter
2023-02-10 18:13:00.00...	All Daily Increm...	IR_Check	64f4538c-af36-4ce8-8645-bf9629454...	Succeeded		CDP1 (West US)	00:56:19.4708658	After
2023-02-09 20:29:00.00...	All Daily Increm...	IR_Check	1de4e778-9be2-4273-8f97-6c4194ba...	Succeeded		CDP1 (West US)	01:10:19.0671877	Current
2023-02-09 19:19:00.00...	All Daily Increm...	IR_Check	556fb137-2b8f-4152-bbaf-cef0dd525...	Failed		CDP1 (West US)	00:41:03.8557894	Before
2023-02-09 18:19:00.00...	All Daily Increm...	IR_Check	87c7a119-05c4-4730-b8f3-a5d6b0f1...	Failed		CDP1 (West US)	00:20:43.5947510	Before
2023-02-09 17:58:00.00...	All Daily Increm...	IR_Check	24a8ed3f-9b04-4915-8d52-50e01320...	Failed		CDP1 (West US)	00:30:04.1093600	Before

Use above information to find if other runs for same activity were successful or not, if you find any successful run, then you have a comparison run to compare the transformedEvents for both runs.

Common Scenarios and explanations :

After finding the last transformedEvent narrow down on following scenarios. You have list of transformations in the transformedEvent which caused the issue. In case logging is turned off by customer, you will not be able to see much of transformation names but still for smaller dataflow with definition of dataflow you should be able to correlate what is going on with the transformedEvents.

Scenario 1 [SQL kind sources]:

All SQL, PostgreSQL, MySQL, and DW (without staging) are being read into a single partition, that is where OOM can happen. To resolve this, we should add source-based partitioning (for DW source by using staging).

Scenario 2 [JSON/CosmosDB Flatten]:

This is generally applicable for hierarchical data sources like JSON, CosmosDB, Office 365, customers would like to flatten their data. When flatten happens, rows like **MyValue1, [1,2,3]** gets converted in to 3 rows as below :

MyValue1, 1

MyValue1, 2

MyValue1, 3

As you see, this results in 3 times data and can result in a lot of data in a single partition, so the customer should increase partitions before such flatten. if earlier default partitions in the previous transformation were 10, make it 50 (round robin).

Scenario 3 [Joins, Aggregate, window, pivot]:

If a lot of data is read in the source like more than 300GB, and then there is join/Exists/Lookup/Windows/Aggregate/Pivot (these transformations shuffle the data to 200 default partitions), then basically all 300GB data is going to 200 partitions making on an average 1.5GB to single partition (IMO max size partitions can take is around 2GB), which is close to limits, joins/exists/windows increase row size, so customers may end up OOM. For these shuffle partitions, there might be a need to be increased the partitions by PG. In this case, get all the required details and provide them in AVA before involving DataflowPG.

Scenario 4 [Delta writer]:

If the user is writing to delta and let's say there are more than 300GB of data in the target. Delta writes by joining the data in the sink with the data to be written to delta, and this may max out the partition size (as mentioned above), which will result in OOM again. For this, there might be a need to change the configuration for now by PG (as PG is going to expose the way for the customer to change these very soon).

Scenario 5 [Union transformation]:

The customer dataflows may intermittently receive an out-of-memory (OOM) error. In this case, the customer is using both debug/activity runs and tried increasing the Core and Memory of the IR. However, failing at **Union** Transformation with the following error.

DF-Executor-InternalServerError java.lang.Exception: Persist failed with out-of-memory error for transformation UnionAll com.microsoft.datafactory.dataflow.common.exception.RuntimeException: java.lang.Exception: Persist failed with out of memory error for transformation UnionAll

In this case, the customer has the following transformations.

Recommendation:

- In the case of a union, if a source is large, and the partition has lots of rows, then doing a union with another stream can run in to out of memory, in that case, you need to ask the customer to repartition (increase partitions) on the source with larger data. Repartitioning needs to be applied on transformation before union not on union. Partitioning is always applied to the result of transformation, so in case a union is failing with OOM and you try repartitioning on union itself that will not work, as result of union was not being calculated earlier as well. So apply repartitioning in incoming streams.
- If you notice the error happened with persist call, you may ask the customer to turn the logging level off in their Dataflow.

- Also recommend customer to use activity run time for debug pipeline.

Explanation

Union merges the streams partitions by partitions. Let's say, there are 3 streams unioned together and all have single partitions like with [100,200, 300] and union with [1000, 2000, 3000] to result in partitions with row count [1100, 2200, 3300], so if 3000 was about limit it will push it over to have out of memory, that's why we are asking to make more partitions and distribute this data. The above numbers are just an example, rows can be in millions. The point is on average it becomes 3 times and leads to OOM.

Let's assume all have single partitions only [100], [200], [300] rows, then the union will result in 600 rows in single partitions, now if from the default of 300 max we suddenly got 600 rows (these are just example, rows can be in millions), the point is on average it becomes 3 times, and that's why it can result into OOM.

Troubleshooting

As mentioned above, please follow the troubleshooting steps and guidelines mentioned in the above wiki and if the problem still persists.

Also, please verify what the customer is mentioning and find out:

- What are the incoming rows from different sources. Please verify with customer again if our logs are saying different numbers, customer may see their datasets differently, like they consider one of the source as main source, but for a run all sources matter.
- What is the Source/Sink
- Check what is the logging level?
- What is the data size.
- Refer to this 2209080040007794 AVA thread for troubleshooting.

How good have you found this content?