

JSON,Excel,XML - OOM Issue

Last updated by | Zhangyi Yu | Apr 22, 2022 at 5:03 AM PDT

Contents

- [Issue](#)
- [Root Cause](#)
- [Resolution](#)
- [Additional Information:](#)

Issue

When customer try to read big JSON/Excel/XML file in Azure Data Factory, he/she meets OOM issue during the activity execution.

Root Cause

The OOM issue of reading big xml file is by design, root cause is that we must read whole xml file into memory while it is single object, then infer schema and get data.

The OOM issue of reading big excel file is by design, root cause is that the SDK (POI/NPOI) we used must read whole excel file into memory, then infer schema and get data.

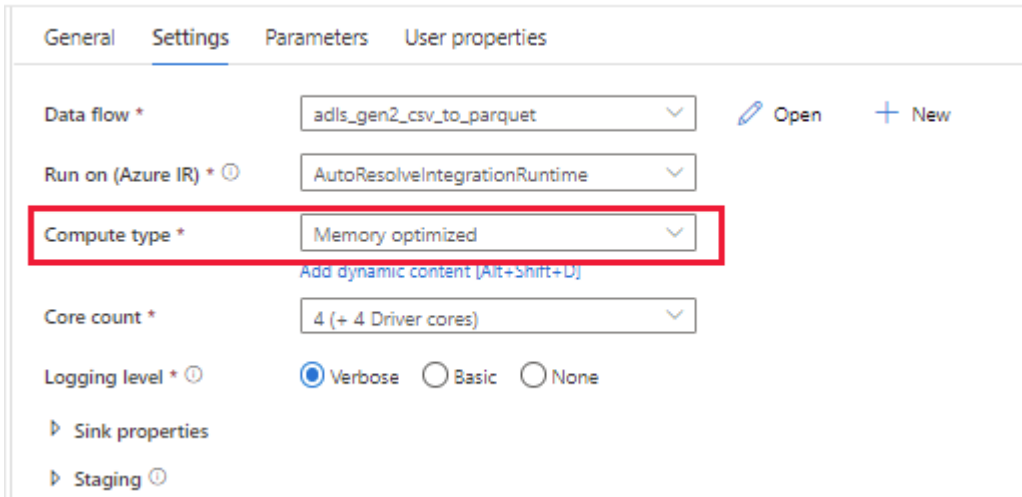
The OOM issue of reading big json file is by design when the JSON file is single object.

Resolution

When customers meet up issue, please ask them to use one of following options to work around it:

Option-1: Register and online self-hosted IR with powerful machine (high CPU/Memory) to read data from the big file through copy activity.

Option-2: Use memory optimized + big size (for example, 48 cores...) cluster to read data from the big file through dataflow activity.



The screenshot shows the 'Settings' tab of an Azure Data Factory activity configuration. The 'Compute type' dropdown is highlighted with a red box and is set to 'Memory optimized'. Below it, there is a link 'Add dynamic content (Alt+Shift+U)'. Other settings include 'Data flow' set to 'adls_gen2_csv_to_parquet', 'Run on (Azure IR)' set to 'AutoResolveIntegrationRuntime', 'Core count' set to '4 (+ 4 Driver cores)', and 'Logging level' set to 'Verbose' (with 'Basic' and 'None' as options). There are also expandable sections for 'Sink properties' and 'Staging'.

Option-3: Split big file into small ones, then use copy or dataflow activity to read the folder.

Option-4: If customer meets stuck or OOM issue during copy XML/Excel/JSON folder, please suggest him to use pipeline "foreach + copy/dataflow" to handle each file or sub-folder.

Option-5: Others

- For XML, you can use notebook activity with memory optimized cluster to read data from file (if each element has the same schema), today spark itself has different implementation to handle XML.
- For JSON, try different document form (single document, document per line, array of documents) under dataflow source with JSON Settings (<https://docs.microsoft.com/en-us/azure/data-factory/format-json#source-format-options>). If the content of JSON file is "Document per line", it just needs to consume few memory.

Additional Information:

- Icm Reference: N/A
- Author: Zhangyi Yu
- Reviewer: Zhuoyang Zhang; Xiaojin Wang