

Ingest to Azure DB Perf check list

Last updated by | Veena Pachauri | Mar 8, 2023 at 11:57 PM PST

Ingest to Azure DB Perf check list

Tuesday, November 13, 2018

5:05 PM

Most common issues:

1. Azure DB DTU usage
 - o The customer can [monitor the performance](#) and check the database transaction unit (DTU) percentage. If it is close to 100%, suggest the customer to scale up the target Azure DB tier for better perf.
 - o FYI, we are using the tier P11 to get the [reference number](#)
2. WriteBatchSize
 - a. Suggest a larger writeBatchSize like **100,000**. By default it is 10,000.
3. DIU and parallelCopies
 - o If the source is a single large file, the max DIUs we can leverage now is 4:
 - Suggest the customer to generate multiple files to achieve better perf.
 - Else, using single file, the suggested setting is **4 DIUs and 4 parallelCopies**.
 - o If the source is a tabular data source, the max DIUs we can leverage now is 4:
 - Suggest the customer to partition the copy activity to multiple activities based on the source query.
 - For single copy activity, the suggested setting is **4 DIUs and 4 parallelCopies**.
 - o If the source is multiple files, suggest to specify a larger DIUs and parallelCopies (e.g. **32 DIUs and 32 parallelCopies**).
4. Target table index
 - o If target table has index, the performance would be slower than the table without index. The suggestion is to remove all indexes before data ingest, and re-create the index after load complete.
5. Stored procedure
 - o Use bulk load instead of stored procedure. Stored procedure is expected to have worse performance than bulk load.
 - o Check whether the customer is using stored procedure:
 - Query: `CustomLogEvent | where ActivityId == "" and TraceMessage == "TransferServiceJobPayloadAfterConvert"`
 - If property "sink.writeProperties.sqlWriterStoredProcedureName" is specified, it means the customer is using stored procedure to load data.
6. Skip faulty row
 - o If there is rowsSkipped telemetry, it means there are incompatible rows and Copy has to handle rows one by one instead of batching by default and led to poor perf.
 - o Suggest the customer to check those incompatible rows and avoid them to have better perf.
 - o CustomLogEvent
 - | where ActivityId contains "Guid"
 - | where TraceMessage == "TransferServiceUpdateSubJobProgress"
 - | project TIMESTAMP, tm= trimlog(Message)
 - | extend SinkSkippedRowCount = extractjson("\$.SinkSkippedRowCount", tm)
 - | project TIMESTAMP, SinkSkippedRowCount, SinkProcessedTotalRow = tolong(extractjson("\$.RowCount", tm)) + tolong(SinkSkippedRowCount) .

Created with Microsoft OneNote 2016.

How good have you found this content?

