Query fails with OOM - high SQLQERESERVATION memory clerk

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:32 AM PST

Contents

Issue

Query fails indicating that there is not enough memory to run.

There is insufficient system memory in resource pool '{CustomRP}' to run this query

Investigation/Analysis

The analysis should start always at the Waits and the memory clerks associated with the memory grants.

Note, check also this wiki article for more details on memory issues troubleshooting. Also check this TSG.

1 - First, on ASC go to **Performance**->**Overview** and check the waits. If you see RESOURCE_SEMAPHORE as a predominant wait, confirms that we have memory pressure.

2 - On ASC go to **Performance**->**Memory** and check **Memory Grant Waiting & Timeouts**. This will give you the query ashes that are waiting on memory allocation.

3 - After this look at **Performance**->**Memory** -> **Top Memory clerk consumption over time**. This will give you an insight of what are the clerks that are consuming more memory. For a full list of the memory clerks and their meaning check Memory Clerk types

If clerk with the highest value is SQLQERESERVATION: "*This memory clerk is used for Memory Grant allocations, that is memory allocated to queries to perform sort and hash operations during query execution*"

In other words, we are in the presence of queries that have high memory grants for sort or hashing operations (hash match, for example).

You can use the query below to find queries with high memory grants. The info provided on point 2 might also point you to the correct query hashes.

```
MonWiQdsExecStats
| where  TIMESTAMP >= datetime({StartTime}) and  TIMESTAMP <= datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and   database_name  =~ "{LogicalDatabaseName}" and AppName
| where is_primary  == 1
| extend MaxQueryMemoryMB=round(max_max_query_memory_pages*8.0/1024,1)
| summarize max(MaxQueryMemoryMB) by query_hash
| top 20 by max_MaxQueryMemoryMB  desc nulls last
```

After getting the query hash, and since the memory clerk is SQLRESERVATION we need to look at the execution plan.

After getting the execution plan, try to understand the execution path, more specifically the Join algorithms used, and Sort operations. This TSG can help you with execution plan analysis.

Mitigation

The mitigation can go in different directions, depending on each case. Listing all possible situations and solutions:

- Outdated statistics that lead to inaccurate memory requests and/or bad choices in execution plan operators. On this case, update statistics.
- a missing index on a table and more data has to be pulled from each table. Check Indexing guidelines to help you determine a good index based on the query.
- a large sort has to be performed. Note that indexes can have the data already sorted, eliminating the need for the Sort on the execution. Indexing guidelines to help you determine a good index based on the query.
- In some cases the query needs to be rewritten to avoid large JOINS and Sorts.
- Scale the Managed Instance to get more memory.

Internal reference

350768115

347636559

How good have you found this content?