Blocking session id -5 (spid -5)

Last updated by | Holger Linke | Sep 26, 2022 at 3:42 AM PDT

Contents

- Issue
- Investigation / Analysis
- Mitigation
- Public Doc References
- Internal References
- Root Cause Classification

Issue

Customers report that queries, transactions or processes are being blocked by session id -5 (or spid -5).

In a specific scenario, the customer was performing the following operations in parallel:

- Deleting old data in a loop, executing `DELETE TOP 50000 FROM dbo.sourcetable WHERE colCreateTime <= 2022-01-31 12:00:00`
- Working on current data through a `SELECT * INTO temptable FROM dbo.sourcetable WHERE colCreateTime > 2022-01-31 12:00:00`

The execution of both statements was very slow and appeared to be stuck. When checking for blocking, they identified that the `SELECT INTO` was blocked with PAGEIOLATCH_SH on session ID "-5". Even after killing the related sessions and restarting the operations, it quickly ended in the same scenario.

In another specific scenario, blocking spid -5 was seen in Transactional Replication, with the Distribution Cleanup job blocking the Log Reader Agent job. This is the same scenario as above though, with the cleanup job executing a `DELETE TOP 5000` in a loop.

Investigation / Analysis

SQL Server reports blocking session IDs as negative integer values to indicate special conditions. See the separate article Negative Blocking Header for information about other negative session IDs/spids.

Blocking `session id -5` was recently added to improve latch visibility. It indicates that the session is waiting on an asynchronous operation to complete. These are commonly I/O operations holding I/O latches. There is no blocking latch owner, as the initiating latch owner is not tracked due to the asynchronous nature of this latch type. Prior to the addition of session id -5, the same session would have shown as blocking session id 0, even though it was still in a wait state.

Depending on the workload, blocking session id -5 can be a common occurrence. But seen by itself, blocking session id -5 does not indicate a performance problem. Refer to Negative Blocking Session Ids (-5 = Latch ANY TASK RELEASOR) for further details.

Regarding the specific scenarios described in Issue above:
A session ID -5 has been seen in situations where the blocking session has caused lock escalation. Lock escalation may occur if the operation is affecting 5000 rows or more. The 5000 is a hard-coded threshold. In the case described here, the DELETE affected a larger rowset and caused lock escalation to a table lock, thus blocking the SELECT INTO.

Mitigation

Although blocking session id -5 itself does not indicate a performance problem, customers open support tickets asking for details about session id - 5 (or negative session IDs in general). They also relate performance issues to session id -5.

Explain to the customer that blocking session id -5 itself does not indicate a performance problem but is an indication that the session is waiting on an asynchronous action to complete.

Regarding the specific scenarios described in Issue above:
For avoiding the lock escalation to a table lock, you can try setting the delete batch size to 4999, just below the internal threshold of 5000 for lock escalations. For example, use `DELETE TOP 4999 FROM dbo.sourcetable` instead of `DELETE TOP 50000 FROM dbo.sourcetable`.
This has proven to be a good workaround for the equivalent situation in Transactional Replication, where the cleanup job uses a default delete batch size of 5000. See article Log Reader and Distribution Cleanup blocking each other for more details.

Regarding general performance issues:
High Data IO is commonly reported amongst others. Start troubleshooting the reported performance issues using ASC, our intenal Wiki links etc. to identify potential causes and provide solutions to the customer. It should be treated as a regular performance issue, and the Product Group should only be engaged when necessary.

Wiki references:
Performance and Query Execution
IO Troubleshooting

Public Doc References

- Negative Blocking Session Ids (-5 = Latch ANY TASK RELEASOR)
- sys.dm_exec_requests (returns information about each request that is executing in SQL Database)
- Latch wait types

Internal References

- IcM: 294299234
- IcM: 308688113
- IcM: 311980999

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
<Root cause path>: /Performance/Waits/Locking/Blocking

How good have you found this content?