Unable expand tables from SSMS

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:28 AM PST

Contents

Issue

The customer cannot expand tables from SSMS. They can connect to the database, but when trying to open any of the sub-nodes in the database object explorer, the request never completes and the nodes never show their content.

Investigation / Analysis

Issue 1: When SQL Server Management Studio (SSMS) tries to expand the database object explorer, it opens a separate connection in the background, reads some information from the metadata and also acquires some transient shared locks for reading the info. If the locks are already acquired by a long-running transaction or one of the queries that it is blocking, SSMS is unable to acquire the lock, hence unable to expand tables.

Check ASC -> SQL Troubleshooter -> Performance -> Blocking & Deadlocking tab which will show you any "Long Running Transactions". This queries the `MonDmTranActiveTransactions` Kusto table. See article Active Transactions for further steps and recommendations towards mitigation.

Alternatively, you can run the following Kusto query to fetch long running transactions:

```
let srv = "servername";
let db = "databasename";
let startTime = datetime(2022-11-22 06:00:00Z);
let endTime = datetime(2022-11-22 23:00:00Z);
let timeRange = ago(7d);
MonDmTranActiveTransactions
 | where  TIMESTAMP >= startTime
 | where  TIMESTAMP <= endTime
// | where  TIMESTAMP >= timeRange
| where LogicalServerName =~ srv
| where user_db_name =~ db
| where session_id != -1
| extend duration_hour = (end_utc_date - transaction_begin_time) / time(1h)
| summarize max_duration_hour = arg_max(duration_hour,
            NodeName,
            session_id,
            transaction_id,
            transaction_begin_time,
            transaction_type,
            transaction_state,
            program_name,
            status
) by LogicalServerName, user_db_name, database_id, accessed_tempdb
| sort by max_duration_hour desc
| limit 500

LogicalServerName  user_db_name  database_id  accessed_tempdb  max_duration_hour  NodeName  session_id  transa
servername         databasename  5            0                17,9038797222222   _DB_20    127         324734
servername         databasename  5            1                6,51536222222222   _DB_20    87          345004
```

Issue 2: On another, rather obscure scenario, the issue was caused by SSMS opening a separate connection in the background. SSMS opens one connection when connecting to the database, then opens additional connections when expanding the object explorer nodes. In that scenario, the additional connection was routed differently from the initial connection to the database and was ultimately blocked by the on-premise firewall or proxy. This would require a network trace to confirm.

Issue 3: It is also possible that the user lacks the necessary permissions.

Mitigation

- See article Active Transactions for recommendations and considerations towards mitigation. The quick solution will be to kill the long running transaction which is the lead blocker, but that might affect other applications and users.
- Ask the customer to open SSMS on a different machine and network environment (on-premise, home network, Azure VM).
- Ask the customer to connect with a different user name (e.g. make sure to have admin permissions, use the SQL Admin account, ask a colleague to try the same).

How good have you found this content?