Manually resolve in doubt elastic transactions

Last updated by | Charlene Wang | Oct 12, 2020 at 7:46 PM PDT

Contents

Issue

Customer encounter below error:

```
detected a DTC in-doubt transaction with UOW  %ls. Please resolve it following the guideline for Troubleshooti
```

All table rows involved in the 'in-doubt' global transaction are being locked and thus they cannot be accessed.

In-doubt global transactions cannot get resolved in a reasonable time due to a couple of reasons:

- Transaction manager cannot be accessed by resource managers;
- Something prevents transaction manager from responding with the outcome of the transaction

Investigation/Analysis

Before trying to resolve manually an global transaction you need the following information:

- LogicalServerName
- AppName (sqlinstance name) of the affected transactions
- NodeName
- Tenant ring of the affected nodes LogicalServerName, Tenant ring and AppName can be found from *Sterling Servers and Databases.xts* view.

Identify all sql instances that have in-doubt global transactions

1. Open XTS -> Open AdHocQueryToBackendInstance.xts view -> Type the logical server name in the Servers tab -> Click on the logical database in the Databases tab -> Click on a replica in the Replicas view

2. Type "SELECT * FROM sys.dm_tran_global_recovery_transactions" (does not apply if upgrade is stuck, look upgrade messages which also represents in doubt transactions) In the Query Text box; This shows all global transactions in recovery on the selected sql instance. If you see entries after running this query that means that there in-doubt global transactions on the instance. Note down uow of all in-doubt transactions. (from errorlog messages which prints UOW GUID)

3. If there are no rows in above recovery view, it does not mean there are no global transactions trying to resolve, there is another case where DB is going upgrade and emitting a message in errorlog which can be checked by below query

   ```
   MonSQLSystemHealth
   | where message contains "SQL Server DB needs to upgrade and detected a GT in-doubt transaction with UOW"
   | where TIMESTAMP > ago(30m)
   ```

   It is continuous loop, so 30 minute should tell us any ongoing recovery transactions.

Mitigation

1. To roll back the global transaction run (UOW is the transaction id obtained in the previous steps): KILL <UOW> WITH ROLLBACK (this will not impact other transactions)

   OR dbcc traceon(3454, -1) after 1 minute, turn it OFF, dbcc traceoff(3454, -1). Never forget to turn this OFF (use dbcc tracestatus to confirm). This is used at instance level so make sure all transactions needs to be aborted.

2. To commit run (UOW is the transaction id obtained in the previous steps):

   ```
   KILL <UOW> WITH COMMIT
   ```

   OR

   ```
   dbcc traceon(3455, -1)
   ```

   Then turn it back OFF after 1 minute. Never forget to turn it back off. This is used at instance level so make sure all transactions needs to be committed.

3. Continuously run "SELECT * FROM sys.dm_tran_global_recovery_transactions" untill you don't see any output. Doesn't apply for DB upgrade scenario where we need to look kusto query if DB is not looping for outcome anymore.

4. In case we have too many distributions to JIT and apply dbcc, we can instead set these traceflags at server level only if all UOWs are ABORTs OR all UOWs are COMMIT (if we have some transactions aborted and some committed, we cannot set instance level above traceflags)

   ```
   MonSQLSystemHealth
   | where message contains "SQL Server DB needs to upgrade and detected a GT in-doubt transaction with UOW"
   | where TIMESTAMP > ago(30m)
   ```

If all above doesn't work, please raise ICM to global transactions team.

Root Cause Classification

How good have you found this content?

🙂 🙁