Blocking on Read Replica

Last updated by | Holger Linke | Sep 26, 2022 at 3:35 AM PDT

Contents

Issue

The customer has been following the recommendations from article Use read-only replicas to offload read-only query workloads and is directing their read-only workload to the secondary instances of their Premium or Business Critical databases.

Intermittently, they encounter one or both of the following issues:

1. The read-only queries are running slowly or are timing out, as if they were being blocked.
2. The queries are failing with one of the following error messages:

error	text
1219	Your session has been disconnected because of a high priority DDL operation.
3947	The transaction was aborted because the secondary compute failed to catch up redo. Retry the transaction.
3961	Snapshot isolation transaction failed in database 'xxxxx' because the object accessed by the statement has been modified by a DDL statement in another concurrent transaction since the start of this transaction. It is disallowed because the metadata is not versioned. A concurrent update to metadata can lead to inconsistency if mixed with snapshot isolation.

Investigation / Analysis

Possible cause 1

Queries running on read-only replicas need to access metadata for the objects referenced in the query (tables, indexes, statistics, etc.). In rare cases, if object metadata is modified on the primary replica while a query holds a

lock on the same object on the read-only replica, the query can block the process that applies changes from the primary replica to the read-only replica. If such a query were to run for a long time, it would cause the read-only replica to be significantly out of sync with the primary replica.

If a long-running query on a read-only replica directly or indirectly causes this kind of blocking, it may be automatically terminated to avoid excessive data latency and potential database availability impact. The session will then receive error 1219 or error 3947.

To identify any replication lag, you can query sys.dm_database_replica_states on the primary and secondary nodes and check the synchronization state and health, log send queue size and rate, redo queue size and rate, and the secondary lag.

Possible cause 2

In rare cases, if a snapshot isolation transaction accesses object metadata that has been modified in another concurrent transaction, it may fail with error 3961. The error occurs because the metadata is not versioned and any change in the metadata can invalidate the versioned scan under snapshot isolation.

Possible cause 3

Besides the specific scenarios mentioned above, it is also possible that the read-only replica has a performance bottleneck. The read-only replica has to execute the replication load that comes in from the primary replica and the read-only query load from user connections, which can reach the capacity limits on busy databases.

Unfortunately we don't have Query Store yet on secondary nodes, therefore only limit information about executed queries is available. You can still use regular DMVs for monitoring the state of the replica and for troubleshooting.

See Monitoring and troubleshooting read-only replicas for a set of DMVs that are commonly used for replica monitoring and troubleshooting.

Possible cause 4

In other rare cases, it has been found that the auto-create statistics had been triggered on the read-only replica. This will acquire a schema lock to perform the creation of statistics and might lead to one of the scenarios described above.

Mitigation

The **scenarios #1 and #2** described above are largely by design, and there are no good options to eliminate such blocking.
Some best practices may help to reduce the impact:

1. Avoid schema changes on the primary replica under high workload.
2. Capture the listed errors and handle them by retrying the transaction.

Regarding **scenario #3**, run through the usual performance troubleshooting steps. Check resource bottlenecks for CPU and I/O on the read-only replica and use the DMVs from Monitoring and troubleshooting read-only replicas.

Regarding **scenario #4**, change the database settings on the primary replica to perform the auto-update statistics asynchronously:

```
ALTER DATABASE AzureDatabaseName SET AUTO_UPDATE_STATISTICS_ASYNC ON
```

How good have you found this content?