Log Reader and Distribution Cleanup blocking each other

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:37 AM PDT

Contents

Issue

The customer noticed that the Log Reader Agent was not progressing and later failing with a timeout error:

```
2022-02-03 11:01:37.031 OLE DB DistLog 'tcp:distributorserver.<dnssuffix>.database.windows.net': exec sp_MShel
2022-02-03 11:06:27.720 Status: 2, code: 1003, text: 'Query timeout expired, Failed Command: sp_MSget_last_tra
2022-02-03 11:06:27.720 Query timeout expired, Failed Command: sp_MSget_last_transaction @publisher_id = 5, @p
2022-02-03 11:06:27.720 Status: 0, code: 22038, text: 'The process could not get the last distributed transact
2022-02-03 11:06:27.720 Status: 0, code: 22037, text: 'The last step did not log any message!'.
2022-02-03 11:06:27.720 Disconnecting from OLE DB DISTOLE 'tcp:faciliaprod-distributor-distributorserver.<dnss
```

When investigating the reason for the issue, it showed that the Log Reader writer thread was blocked by the Distribution cleanup Job.

Also, after stopping and restarting the Distribution cleanup job, the cleanup was at first blocked by the Log Reader Agent. This however resolved itself after some time by switching head blockers, so that the Distribution cleanup job was again blocking the Log Reader Agent.

The Distribution cleanup overall executed for a long time and had a very low cleanup rate, like this:

```
Executed as user: DB4C1\WF-oM3QjL6VozKSGhF.
Deleted 0 row(s) per millisecond from MSrepl_commands [SQLSTATE 01000] (Message 22121)
Deleted 0 row(s) per millisecond from MSrepl_transactions [SQLSTATE 01000] (Message 22121)
Removed 17724 replicated transactions consisting of 42008 statements in 11009050 milliseconds (0 rows/millisec
The step succeeded.

Executed as user: DB4C1\WF-oM3QjL6VozKSGhF.
Deleted 0 row(s) per millisecond from MSrepl_commands [SQLSTATE 01000] (Message 22121)
Deleted 0 row(s) per millisecond from MSrepl_transactions [SQLSTATE 01000] (Message 22121)
Removed 16716 replicated transactions consisting of 38767 statements in 530717 milliseconds (0 rows/millisec).
The step succeeded.
```

It summarizes into two main symptoms:

- The Log Reader Agent falling behind and the transaction log of the Publisher database growing
- The Distribution cleanup falling behind and the Distribution database growing

Investigation

To check for progress and blocking on the replication agents, you can run the following query at the Distributor:

```
SELECT req.database_id,
   db_name(req.database_id) as DBName,
   req.start_time as 'Start Time',
   convert(varchar, getdate() - req.start_time, 114) as 'Elapsed Time',
   req.cpu_time as 'CPU Time',
   req.dop as 'Parallelism',
   req.wait_type as 'Wait Type',
   req.wait_resource as 'Wait Resource',
   req.session_id as 'Session ID',
   req.blocking_session_id as 'Blocked By',
   ST.text as 'Batch'
   , substring (REPLACE (REPLACE (SUBSTRING (ST.text, (req.statement_start_offset/2) + 1,
   ((CASE statement_end_offset WHEN -1 THEN DATALENGTH(ST.text) ELSE req.statement_end_offset END - req.statem
   , QP.query_plan as 'Execution Plan'
--       , req.*, ST.*
FROM sys.dm_exec_requests AS req
CROSS APPLY sys.dm_exec_sql_text(req.sql_handle) AS ST
CROSS APPLY sys.dm_exec_query_plan(req.plan_handle) AS QP
```

The Log Reader Agent is blocked in the Distribution database when running the stored procedure `sp_MSget_last_transaction` . The blocked statement very likely is the following:

```
select top 1 @max_xact_id = rt.xact_id, @max_xact_seqno = rt.xact_seqno
  from MSrepl_transactions rt
  where
         rt.publisher_database_id = @publisher_database_id and
         not xact_id = 0x0
 order by xact_seqno desc
```

For the Distribution cleanup, the stored procedure `sp_MSDelete_publisherdb_trans` keeps running with the following symptoms:

- The waittype is "PREEMPTIVE_HTTP_REQUEST"
- It is blocked by session ID/spid "-5"
- The current statement is "fetch hCdirs into @dir, @xact_seqno, @command_id, @type"

DBName	RunTime	time_running	batch	current_statement	cpu_time	parallelism	wait_type	wait_resource	session_id	blocked
distribution	06:17.7	25:41.3	CREATE PROCEDURE sp_MSdelete_publisherdb_trans @publisher_database_id int, @max_xact_seqno	fetch hCdirs into @dir, @xact_seqno, @command_id, @type	5901248	1	PAGEIOLATCH_SH	16:1:232361961 (LATCH 0x000001757681E3D8: Class: BUFFER KP: 0 SH: 0 UP: 0 EX: 1 DT: 0 Sublatch: 0 HasWaiters: 1 Task: 0x00000130B435B088 AnyReleasor: 1)	313	-5

Analysis

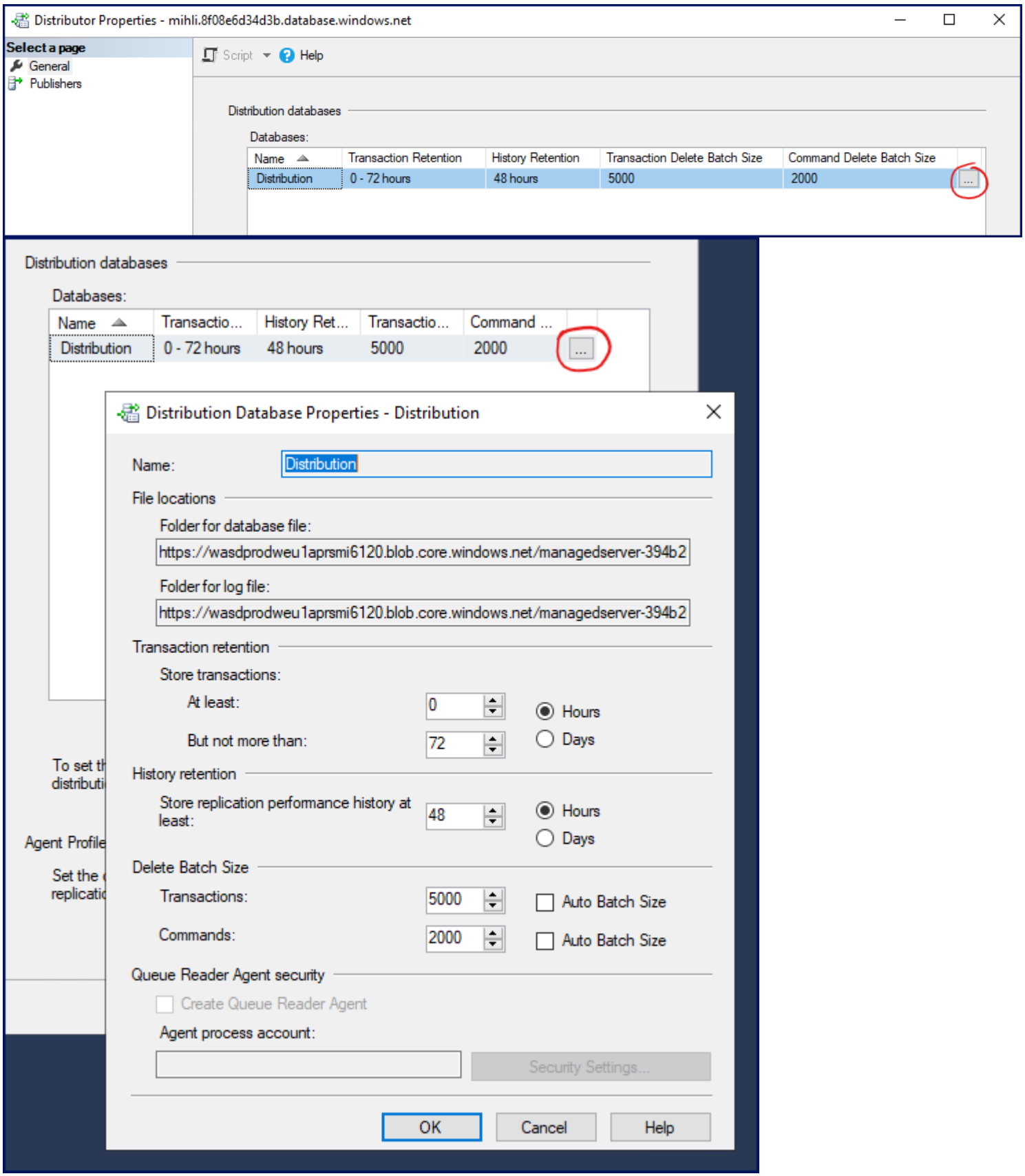
When checking the lock DMV, it showed a table lock on MSrepl_transactions, which indicates lock escalation. Lock escalation typically occurs if the affected number of rows is equal or above 5000 rows.

The default delete batch size for transactions however is exactly 5000 and therefore might cause lock escalation on the MSrepl_transactions system table.

Mitigation

For avoiding the table lock, you can try setting the delete batch size for transactions to 4999, just below the threshold for lock escalations.

You can see and change the current delete batch size on the "Distribution Database Properties" page:



Or you can run the following query at the Distributor:

```
select deletebatchsize_cmd, deletebatchsize_xact from msdb.dbo.Msdistributiondbs
```

To change the delete batch size for transactions, you can either use the Distribution Database Properties dialog as shown above (change Transactions from 5000 to 4999).
Or you can execute the following SQL script to check and change:

```
begin transaction
select deletebatchsize_cmd, deletebatchsize_xact from msdb.dbo.MSdistributiondbs
update msdb.dbo.MSdistributiondbs set deletebatchsize_xact = 4999
select deletebatchsize_cmd, deletebatchsize_xact from msdb.dbo.MSdistributiondbs
commit
```

Internal Reference

IcM 286971791
IcM 299167849

How good have you found this content?