Performance Troubleshooting Initial Steps

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

Contents

Issue

The customer is reporting replication latency and requesting assistance with narrowing down the exact cause.

There are a several possible causes for it, of which one or more can apply at the same time:

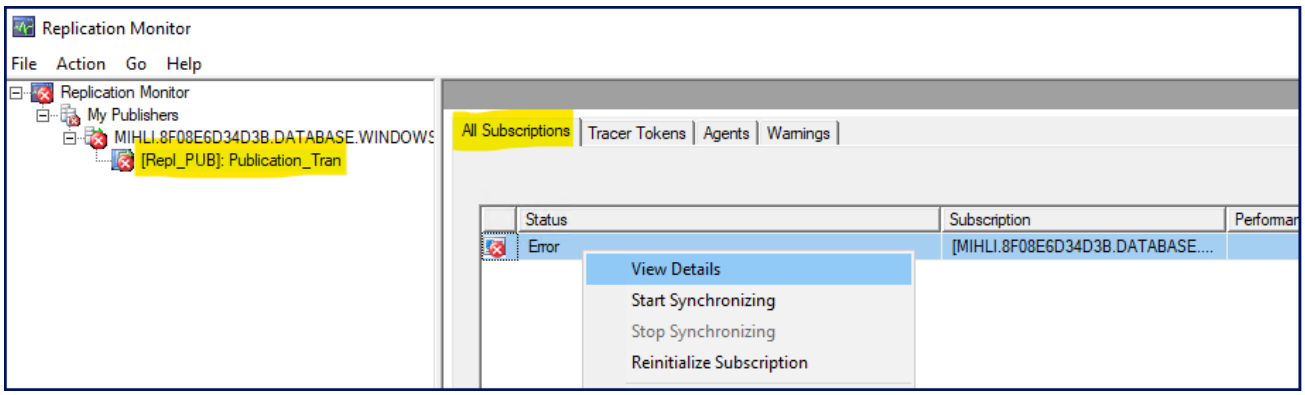
- Log Reader slow to read data changes from the transaction log of the Publisher database
- Log Reader slow to write data changes into the distribution database
- Distribution agent slow to read data changes from the distribution database
- Distribution agent slow to write data changes into the Subscriber database

This article helps you identifying which of the symptoms apply to the customer incident.

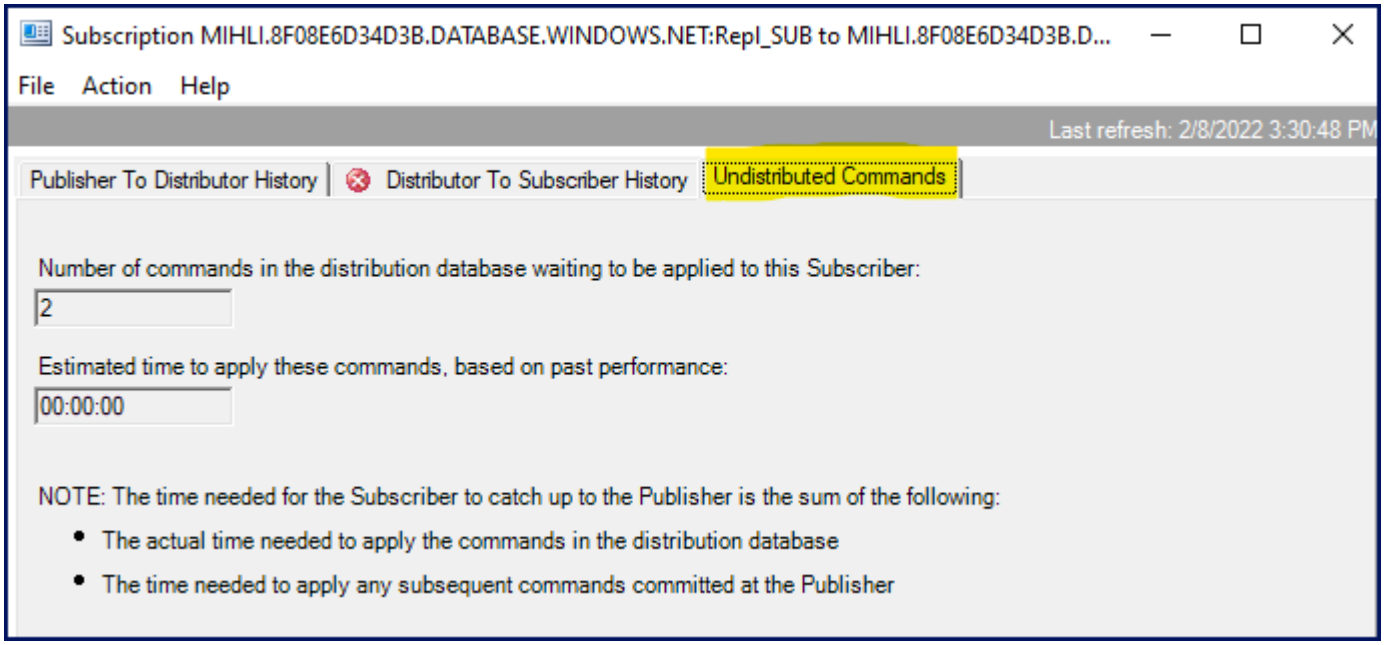
Investigation/Analysis

Replication Monitor: check Subscription - Undistributed commands

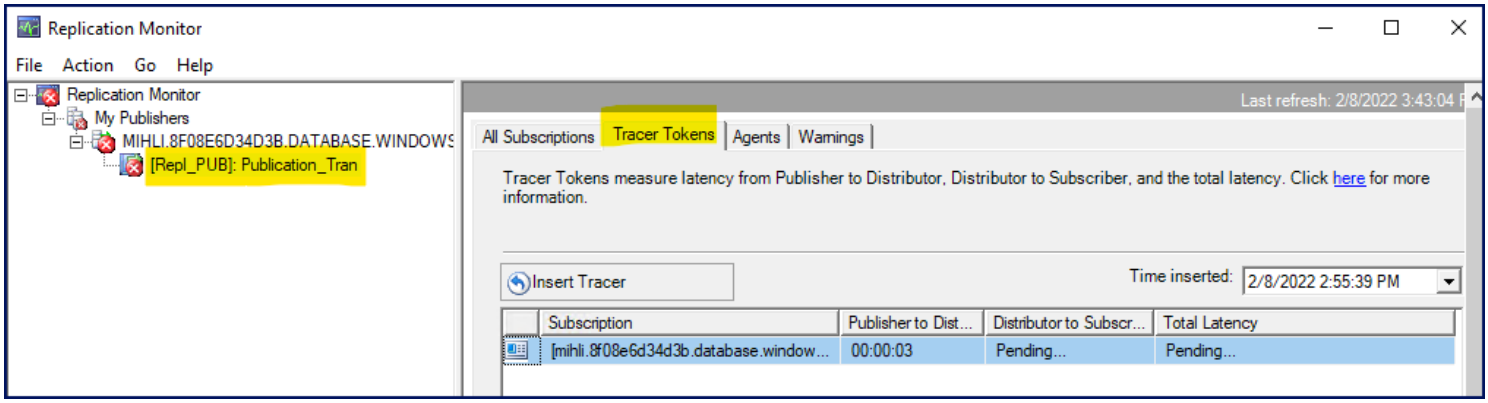
1. Check for failing Distribution agents, or identify the subscription named by the customer.
   Cross-check the "Subscription" column to confirm that you are looking at the correct Subscriber database and subscription:



2. View details for the selected Distribution agent and select the `Undistributed Commands` tab:



Replication Monitor: insert tracer token



A tracer token is a small amount of data that is written to the transaction log of the publication database. It is marked for replication, just as it were a normal replicated transaction, and sent through the system. It allows to calculate the Publisher-to-Distributor and Distributor-to-Subscriber latency.

Guidance:

- If the tracer token does not arrive at the Distributor, the issue is with the Log Reader Agent;
- If it made it to the Distributor but not to the Subscriber, you need to look at the Distribution Agent.

(In the screenshot above, the Log Reader writer thread was blocked in the distribution database, thus the token couldn't be moved to the Distributor)

More information: Tracer Tokens

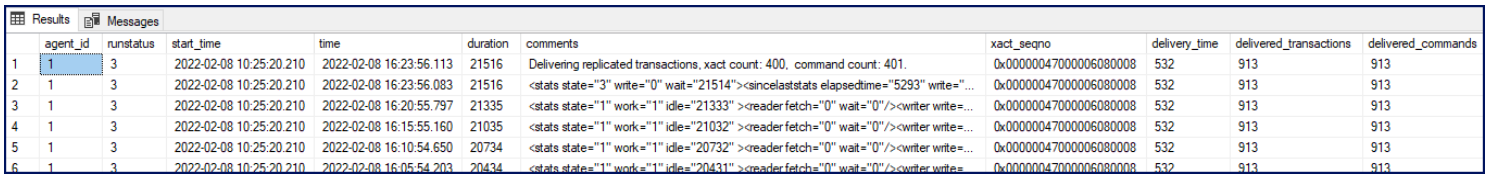
Distribution Database - Log Reader and Distribution Agent history

The replication agents are reporting performance statistics in their history tables. You can check current and past messages through the following queries:

```
-- identify the agent ID from the agent table
select * from Distribution..MSlogreader_agents -- identify agent by checking column: publisher_db
select * from Distribution..MSdistribution_agents -- identify agent by checking columns: publisher_db, subscri

-- set the agent_id that you have identified on the MSxxxx_agents query above
select top 50 * from Distribution..MSlogreader_history /* where agent_id = 1 */ order by start_time desc, time
select top 50 * from Distribution..MSdistribution_history /* where agent_id = 3 */ order by start_time desc, t
```

Sample output - Log Reader history:



Results | Messages

	agent_id	runstatus	start_time	time	duration	comments	xact_seqno	delivery_time	delivered_transactions	delivered_commands
1	1	3	2022-02-08 10:25:20.210	2022-02-08 16:23:56.113	21516	Delivering replicated transactions, xact count: 400, command count: 401.	0x00000047000006080008	532	913	913
2	1	3	2022-02-08 10:25:20.210	2022-02-08 16:23:56.083	21516	<stats state="3" write="0" wait="21514"><sincelaststats elapsedtime="5293" write="...	0x00000047000006080008	532	913	913
3	1	3	2022-02-08 10:25:20.210	2022-02-08 16:20:55.797	21335	<stats state="1" work="1" idle="21333" ><reader fetch="0" wait="0"/><writer write=...	0x00000047000006080008	532	913	913
4	1	3	2022-02-08 10:25:20.210	2022-02-08 16:15:55.160	21035	<stats state="1" work="1" idle="21032" ><reader fetch="0" wait="0"/><writer write=...	0x00000047000006080008	532	913	913
5	1	3	2022-02-08 10:25:20.210	2022-02-08 16:10:54.650	20734	<stats state="1" work="1" idle="20732" ><reader fetch="0" wait="0"/><writer write=...	0x00000047000006080008	532	913	913
6	1	3	2022-02-08 10:25:20.210	2022-02-08 16:05:54.203	20434	<stats state="1" work="1" idle="20431" ><reader fetch="0" wait="0"/><writer write=	0x00000047000006080008	532	913	913

```
 <stats state="3" write="0" wait="21514"><sincelaststats elapsedtime="5293" write="0" wait="21514"/>
<message>Raised events that are generated only by the Log Reader Agent when the writer thread waits longer than
the -messageinterval time. If you notice State 3 events that are recorded for the Log Reader Agent, this
indicates that the agent is taking a long time to scan the replicated changes from the transaction log.
</message></stats>

 <stats state="1" work="846" idle="21513" ><reader fetch="0" wait="0"/><writer write="846" wait="21514"/>
<sincelaststats elapsedtime="1026" work="845" cmds="500" cmdspersec="0.000000"><reader fetch="0" wait="0"/>
<writer write="845" wait="5293"/></sincelaststats><message>Normal events that describe both the reader and
writer thread performance.</message></stats>
```

Note the explanation in the <message></message> section for further insights. (In this specific situation, the log reader writer thread was blocked in the distribution database, and the history writer could not progress further to show the correct wait reason)

This topic is discussed in further detail at: [Performance: Log Reader and Distribution Agent Statistics]

Check if Replication Agents are blocked by other processes

The replication agents are using SQL connections to perform their various tasks:

- The Log Reader Agent is writing into the Distribution database (data, metadata, and history)
- The Distribution Agent is reading from the Distribution database (data)
- The Distribution Agent is writing into the Distribution database (history)
- The Distribution Agent is writing into the Subscriber database (data and metadata)

Either of these connections might get blocked by other users or tasks:

- on the distribution database by other replication agents or by the replication cleanup tasks
- on the Subscriber database by user activity (applications reading data, user ad-hoc queries)

It makes therefore sense to check for blocking on the Distributor and Subscriber, especially for scenarios where replications appear to be stuck or running into timeouts.

Use the following query, or your own favourite query, to identify if replication agents are involved in blocking:

```
select spid, blocked, waittype, waittime, lastwaittype, waitresource, open_tran, db_name(dbid), uid, cpu, phys
   memusage, login_time, last_batch, status, hostname, program_name, cmd, loginame
from sysprocesses
```

Check the `blocked` column to identify blocked process and look in the `program_name` column for agent names:

- Log Reader Agents named like: Repl-LogRreader-0-xxxxx
- Distribution Agents named like: servername_publisherdb_publicationname

Example names:

> mihli.8f08e6d34_Repl_PUB_Publication_Tran (*writer thread to subscriber - might be more than 1 spid*)
> mihli.8f08e6d34d3b.da-Repl_PUB-Publication_Tran-mihli.8f08e6d34d3b.da-3 (*reader thread to distribution database*)

Narrow down on database names and check if any of the replication agent spids are blocked.

Mitigation

If you have identified which replication agent and operation is affected - then continue the troubleshooting with the corresponding section in article:
Performance: Troubleshooting Log Reader and Distribution Agent Performance

How good have you found this content?