Performance: Log Reader and Distribution Agent Statistics

Last updated by | Holger Linke | Apr 22, 2022 at 8:59 AM PDT

Contents

Issue

The customer is suspecting that there is a bottleneck on their Transactional Replication workflow. The ask is to get a quick set of performance statistics for the replication agents that can be monitored over time.

Investigation

The performance statistics for each replication agent are recorded in the history tables for the Log Reader Agent and for the Distribution Agent every five minutes. You may see these statistics in the agent history tables in the Distribution database:

```
select top 1000 comments, * from Distribution..MSlogreader_history order by start_time desc, time desc
select top 1000 comments, * from Distribution..MSdistribution_history order by start_time desc, time desc
```

There are three state events that can be recorded:

State	Description
1	Normal events that describe both the reader and writer thread performance.
2	Raised events that occur when an agent's reader thread waits longer than the agent's -messageinterval time. (By default, the time is 60 seconds.) If you notice State 2 events that are recorded for an agent, this indicates that the agent is taking a long time to write changes to the destination.
3	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.

The following is a sample output for the Log Reader Agent:

```
<stats state="1" work="9" idle="288760" >
  <reader fetch="1" wait="0"/>
  <writer write="0" wait="1632"/>
  <sincelaststats elapsedtime="300" work="9" cmds="0" cmdspersec="0.000000">
    <reader fetch="8" wait="0"/>
    <writer write="9" wait="0"/>
  </sincelaststats>
</stats>
```

There are several things that you can see from this output:

- The first three lines give you a summary for the period since the agent had been started. The values are measured in seconds.
- You can see there that the agent had been idle during most of its execution (work=9 very low, idle=288760 relatively high); these are cumulative values while the agent had been running.
- The low cumulative value for `reader fetch` indicates that there was almost no data changes since the agent had been started.
- The `writer wait="1632"` indicates that there were times when the writer thread had to wait for the reader thread to retrieve changes from the transaction log.
- The `sincelaststats` section shows you the metrics for the past 300 seconds = 5 minutes. There were minimal work time and no commands.
- The low values for fetch, write, and wait confirm that the agent was idle during the past 5 minutes.

By default, the historical data is retained only for the last 48 hours. The `Agent history clean up: Distribution` job will remove data that is older than 48 hours. This default value can be changed through the sp_changedistributiondb stored procedure by specifying a new value for the history_retention parameter.

Analysis

Example: Distribution Agent Reader Thread

This example demonstrates a situation with replication latency where the bottleneck is the Distribution Agent reader thread. The reader thread queries the Distribution database for commands to apply at the subscriber.

```
<stats state="1" work="14798" idle="2035">
  <reader fetch="14798" wait="193"/>
  <writer write="12373" wait="9888"/>
  <sincelaststats elapsedtime="424" work="415" cmds="296900" cmdspersec="713.000000">
    <reader fetch="415" wait="7"/>
    <writer write="377" wait="212"/>
  </sincelaststats>
</stats>
```

There are again several things that you can see from this output:

- For sincelaststats, the elapsed time is 424 seconds, of which 415 seconds are work time. This means that the agent was very busy, which is also reflected in the counters for commands and commands per second.
- The `sincelaststats..reader fetch` is equal to the overall work time of 415 seconds, which indicates that the agent was 100% busy on this thread; this is also confirmed through the very low `wait` value.
- The `sincelaststats..writer write` is high, but lower than the overall elapsed time; the `wait` is half of the elapsed time, indicating that the writer thread was waiting to receive data from the reader thread for applying it to the Subscriber database.

If you observe high writer wait times combined with high reader fetch times, then investigate the performance of the Distribution Agent execution against the Distribution database.

Example: Distribution Agent Writer Thread

This example demonstrates a situation with replication latency where the bottleneck is the Distribution Agent writer thread. The writer thread applies the replicated commands to the Subscriber database.

Note that the state is 2, and the output is somewhat different than the state=1 statistics:

```
<stats state="2" fetch="48" wait="384" cmds="1028" callstogetreplcmds="321">
  <sincelaststats elapsedtime="312" fetch="47" wait="284" cmds="1028" cmdspersec="3.000000"/>
</stats>
```

State=2 indicates that the reader thread had to wait longer than the configured `-messageinterval`. The `-messageinterval` is a configuration parameter of the replication agents, with a default of 60 seconds.

If `-messageinterval` is increased (in the output below it was set to 240 seconds), you may once again receive state=1 statistics. Note how the sincelaststats fetch wait time is very high, at 505 seconds, and writer write time is close or equal to the elapsedtime and work values:

```
<stats state="1" work="1941" idle="0">
  <reader fetch="717" wait="1225"/>
  <writer write="1941" wait="134"/>
  <sincelaststats elapsedtime="764" work="764" cmds="1170730" cmdspersec="1530.000000">
    <reader fetch="258" wait="505"/>
    <writer write="764" wait="50"/>
  </sincelaststats>
</stats>
```

If you observe high reader wait times combined with high writer write times, then investigate the performance of the Distribution Agent execution against the Subscriber database.

Example: Log Reader Agent Reader Thread

This example demonstrates a situation with replication latency where the bottleneck is the Log Reader Agent reader thread. The Log Reader Agent reader thread scans the Publisher database transaction log for commands to deliver to the Distribution database.

```
<stats state="1" work="301" idle="0" >
  <reader fetch="278" wait="0"/>
  <writer write="12" wait="288"/>
  <sincelaststats elapsedtime="301" work="301" cmds="104500" cmdspersec="347.000000">
    <reader fetch="278" wait="0"/>
    <writer write="12" wait="288"/>
  </sincelaststats>
</stats>
```

The sincelaststats writer wait time appears high, at 288 seconds of wait time. This is time that the writer thread is waiting for the reader thread to supply buffers to apply. The reader fetch is also close to the elapsedtime and work, indicating that the reader thread was busy most of the time.

From this output, you can also conclude that restarting the Log Reader Agent has not resolved the issue: the summary values are identical to the sincelaststats values, so this is the first iteration after the agent start.

If you observe high writer wait threads combined with high reader fetch times, then investigate the performance of the Log Reader Agent execution against the Publisher database.

Mitigation

Refer to the performance troubleshooting TSGs for more information on mitigation steps.

Specifically take a look at the scenarios discussed in Performance: Troubleshooting Log Reader and Distribution Agent Performance.

More Information

The following is a line by line description of what each performance statistic is reporting.
Source: Introduction to the performance statistics tools for Replication Log Reader and Replication Distribution agents

Statistic	**State**	**Description**
State		State 1: indicates that it is a normal performance report after a batch commit. State 2: Reader Thread indicates that a batch read waits longer time than the messageinterval property. State 3 Writer Thread indicates that a batch write waits longer time than the messageinterval property.
cmds	2 Only	Indicates the number of commands read by the Distribution agent.
callstogetreplcmds	2 Only	Indicates the number of calls to procedure sp_MSget_repl_commands by the Distribution agent.
work		The value represents the cumulative time that the agent spent on work since the last agent start. The time excludes the idle time.
idle		The value represents the cumulative time that the agent waits to call the sp_replcmds stored procedure when the previous call returns no transactions or when the number of the transactions is smaller than the MaxTrans property since the last agent start
reader fetch		The value represents the cumulative time that the reader spent since the last agent start. The time excludes the idle time and the wait-for-writer time.
reader wait		The value represents the cumulative wait-for-writer time since the last agent start. The value shows the time that is spent waiting for the writer thread to finish consuming data buffer before the reader can fill them again.
writer write		The value represents the cumulative time that the writer spent since the last agent start. The time excludes the idle time and the wait-for-reader time. writer wait The value represents the wait-for-reader time since the last agent start. The value shows the time that is spent waiting for the reader thread to finish populating data buffer before the writer can apply them.

sincelaststats_elapsed_time		The sincelaststats node shows similar statistics for the period beginning at the last recorded stats event. By default, the period is five minutes. The time excludes the idle time. The value represents the time elapsed since last recorded stats event.
sincelaststats work		The value represents the time that the agent spent since last stats event.
sincelaststats cmds		The value represents the number of commands since the last stats event.
sincelaststats cmdspersec		The value represents the number of commands that are performed per second since the last stats event.
sincelaststats\reader fetch		The value represents the cumulative time that the reader spent since the last stats event. The time excludes the idle time and the wait-for-writer time.
sincelaststats\reader wait		The value represents the cumulative wait-for-writer time since the last stats event. The value shows the time that is spent waiting for the writer thread to finish consuming data buffer before reader can fill them again.
sincelaststats\writer		The value represents the cumulative time that writer spent since the last stats event. The time excludes the idle time and the wait-for-reader time.
sincelaststats\writer wait		The value represents the wait-for-reader time since last stats event. The value shows the time that is spent waiting for the reader thread to finish populating data buffer before writer can apply them.

Pubic Doc Reference

Introduction to the performance statistics tools for Replication Log Reader and Replication Distribution agents

How good have you found this content?