Managing Latency in NonStop Systems

Reducing Replication and Application Latency

One of the most critical aspects of data replication performance is latency, and the two most important types are *replication latency* and *application latency*. This brief will focus on these metrics in NonStop to NonStop replication architectures, using both asynchronous and synchronous replication technology.

Replication Latency

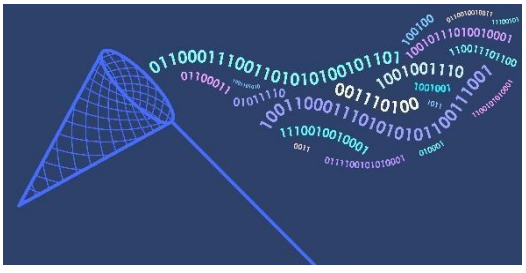
Replication latency is the elapsed time between when a data change (e.g., an insert, update, or delete event) either occurs at the source or is committed to the source system database, to when it is applied to the target database. We say ***either*** occurs or is committed as not all data changes materialize and are made available to the data replication engine when they actually occur (for example, if they are held in cache). In many cases, the changes are only made available some time after the I/O occurred at the source, perhaps due to flushing or a commit operation durability step. Hence, the first availability of the change data to the replication engine marks the time used for when the change occurred.

In a practical sense, replication latency thus starts at this time, and covers the time period when the change data is in the replication pipeline. It is affected by such things as system utilization, process priorities, and network bandwidth. This time period is very important because it represents source production change data that is at risk – it is not yet backed-up, and may be lost in the event of an outageof the source or target system. The longer this time interval is, the more data may be lost.[1]

Recovery Point Objective (RPO)

For this reason, companies typically assign a value for this metric, also known as the RPO. This is a time period which the replication latency cannot exceed in order to minimize (or eliminate) data loss and meet required Service Level Agreements (SLAs). In general, there is a trade-off between replication latency and replication efficiency; the better (lower) the replication latency,the greater the replication overhead (increased system utilization), and vice versa.

Application Latency



Application latency is primarily applicable when using synchronous replication, for example the Shadowbase ZeroData Loss (ZDL) solution. When the source application calls transaction commit to make its data changes permanent, Shadowbase ZDL must ensure that the data changes are safe-stored on the target system (either in memory, in a queue file, or in the target database, depending on the configuration), before allowing the source transaction to commit (if not, then the source transaction outcome depends upon the configuration).[2] In this way, Shadowbase ZDL ensures that no data committed at the source will be lost in the event of a source system outage (RPO =0). However, a consequence of this process is that the time taken for the source application's transaction commit tocomplete may be extended. This additional commit time is known as the *application latency*.[3]

[1]In the case of an asynchronous active/active replication architecture, this time period also represents the time window when a data collision is possible if the same data record is concurrently changed on the source and target system. For more information, see www.shadowbasesoftware.com/publications/white-papers/active-active-technology/.
[2]Shadowbase ZDL provides various options to the customer on how to proceed in the event that data cannot be replicated synchronously.
[3]While replication latency is a contributing factor towards application latency, it does not account for all of it.

The potential impacts of increased application latency include increased transaction response times, longer source record lockduration, a larger number of concurrent transactions, and an increase in overall system utilization. For these reasons, when using synchronous replication, application latency SLA specifications must be set which should not be exceeded.

In order to help ensure SLA metrics for both replication and application latency are being met, as well as raising alerts when they are not, Shadowbase software includes various options to control, monitor, notify, and set alarms for these metrics at various points during the replication process.

Managing Replication Latency (NonStop to NonStop)

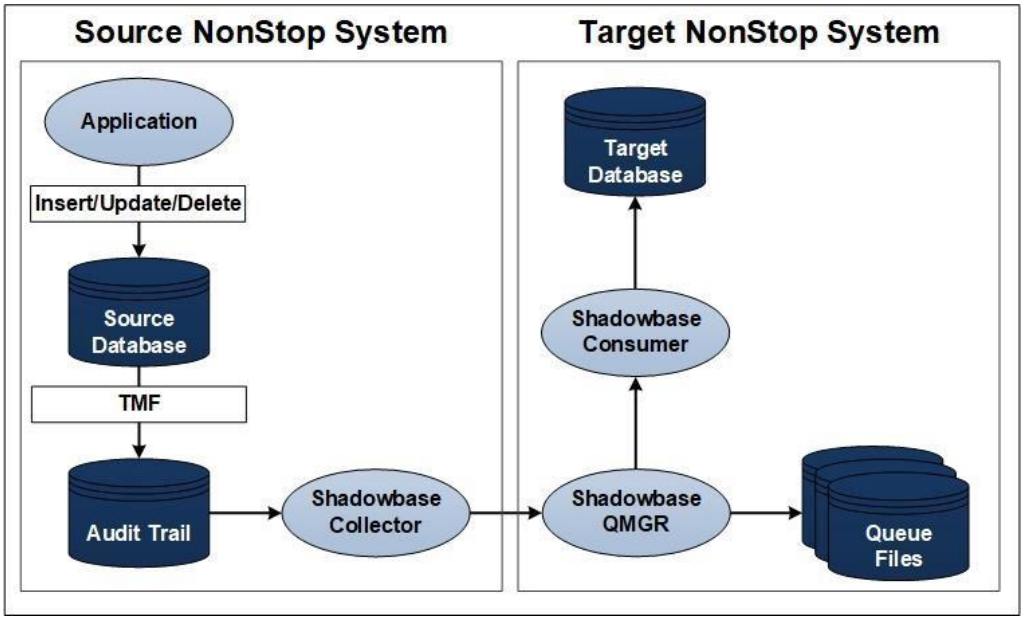


Figure 1 – Typical HPE NonStop Shadowbase Active/Passive Data Replication Configuration

Figure 1 shows a typical active/passive HPE NonStop Shadowbase uni-directional source and target data replication configuration.[4] On the active system, the Source NonStop system, an application makes changes to a source database table/file (Enscribe, SQL/MP, SQL/MX). These changes are committed to the database via the TMF audit trail, and the Shadowbase Collectorprocess reads these changes from the audit trail and sends them across the network to the target system (using either Expand or TCP/IP communication).

On the target system, the Target NonStop system, the Shadowbase Queue Manager (QMGR) process receives these changes, safe-stores them either in memory or to persistent disk, and acknowledges receipt of the message back to the Collector. Inparallel, the QMGR forwards the events to the Consumer to apply the changes into the target database. Replication latency is represented by the time between the data changes appearing in the TMF audit trail on the source system (because that is when they materialize, or are made available to the Shadowbase Collector), and their safe-store in the Shadowbase QMGR/Queue Files, or subsequent application into the database on the target system.[5]

To monitor this replication latency, it is possible to set Shadowbase LATENCYTHRESHOLD values for both the Collector and Consumer processes. If the latency of either a Collector or Consumer process falls outside of these values, Shadowbase software will issue a warning message. LATENCYTHRESHOLD values should be set equivalent to the desired maximum replication latency. When LATENCYTHRESHOLD warnings are triggered, it is then possible to drill-down further to determine the cause of the situation and resolve it (e.g., how long the lagging condition has been present, and *how far* behind). Once a process is back within its LATENCYTHRESHOLD limit, a message is issued.

[4]While the configuration details may be different, the latency monitoring and management facilities are equally applicable to a heterogeneous replication environment involving other platforms and databases.
[5]Replication latency measures the time from when the I/O materializes in the audit trail (is made available to be replicated), not when the I/O was applied by the application into the source database. On an HPE NonStop system, TMF flushing of the change data from the data disk process to the audit disk process into the audit trail isindeterminate and will vary, and the file system does not record nor save the actual time when the application executed the I/O.

Calculating Source and Target Replication Latencies

The overall replication latency is a product of both the latency on the source (read/send) side, and on the target (receive/applier) side. Shadowbase software provides means to determine both source and target replication latencies independently, as well as overall.

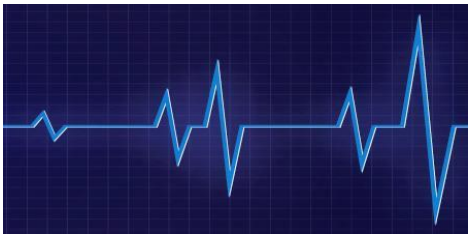
To calculate the source Collector latency:

1. The Collector reads the TMF audit trail and saves the timestamp of the oldest read event.
2. The Collector sends the data change messages to the Consumer; all messages not subsequently acknowledged by the Consumer are marked "in-flight."
3. Periodically, the Collector compares the oldest event timestamp of all the "in-flight" messages to the current system clock.[6]
4. If the time difference is greater than the Collector's LATENCYTHRESHOLD value, a "Collector is behind" message is issued; these messages are then reissued periodically until the Collector is back below its latency threshold.

To calculate the target Consumer latency:

1. The Consumer receives the timestamp of each event in the messages from the Collector.
2. After it processes an event it compares the timestamp of the event to the current system clock.
3. If the difference is greater than the Consumer's LATENCYTHRESHOLD value, a "Consumer is behind" message is issued; these messages are then reissued periodically until the Consumer is back below its latency threshold.

Pulse Feature



Shadowbase software also includes a pulse feature to measure end-to-end replication latency, assisting those customers that need to adhere to a strict service-level agreement (SLA). This feature is used to measure the amount of time that an inserted "pulse" record takes to traverse from the source database, through the replication process, to the target database (just like a "regular" application database change). Results are returned to the source system and stored in a pulse file (allowing review of historical pulse timings). In addition, warning messages are issued when a completed pulse event exceeds a pre-defined threshold value or times-out (i.e., no response is received).

Pulse records can be automatically generated based upon a pre-defined interval (e.g., every 60 seconds), and/or can be generated interactively on-demand via an operator command. Therefore, the pulse feature can be used as needed to verify that replication latency remains within prescribed limits.

If an unacceptable level of replication latency is detected which does not resolve, there are numerous tuning and configuration parameters provided by Shadowbase software which can help lower it below the required threshold. For more details of these parameters and those discussed above, please consult the *Shadowbase Operations Manual*.

Managing Application Latency

As discussed above, synchronous replication also introduces the concept of application latency (as well as replication latency, which still applies).[7] As is the case with replication latency, there is a trade-off between application latency and replication efficiency. While Shadowbase ZDL provides numerous optimizations and control parameters to minimize application latency, it still needs to be managed in order to ensure it does not significantly impact source application transaction response times and violate SLAs.

When moving to a synchronous replication environment, it is necessary to proceed in an incremental fashion, in order to assess and minimize the potential impact of application latency on the application, *before* putting the synchronous replication environment into production. The following series of steps are recommended.

[6] Because replication latency is measured using system clocks as a basis, it is recommended to utilize a universal time synchronization product to keep the clocks synchronized between source and target systems, thereby ensuring the accuracy of the latency calculations.
[7] Note that application latency exists even in an asynchronous replication environment, but it is unaffected by it. (Asynchronous replication is decoupled from the source application's transaction and does not participate in the transaction completion process.)
[8] Shadowbase synchronous replication is built upon Shadowbase asynchronous replication, so it is imperative to first create a well-tuned asynchronous replication environment.

Step One: Configure and Tune a Shadowbase Asynchronous Replication Environment

Before putting a Shadowbase *synchronous* environment into production, the first step is to configure a Shadowbase *asynchronous* replication environment and tune it for the expected normal and peak application loads.[8] Next, run the environment in a special Shadowbase *synchronous monitor mode*, where transactions are replicated asynchronously and no additional application latency is introduced; however, the transactions are tracked and monitored as if they were synchronous.

Statistics generated for these transactions are collected and displayed, allowing the measurement of the potential impact of synchronous replication with minimal impact on current operations ("what-if analysis"). Using this approach, synchronous monitor mode obtains a close approximation of the potential application latency, without actually incurringany additional application latency.

Step Two: Specify Synchronous Replication for Necessary Transactions

The second step to managing application latency is to only specify synchronous replication for those transactions that really need it. Not all data is created equal – the potential loss of some low-value data may be acceptable, whereas other data is highly valuable, and no loss can be tolerated. Shadowbase ZDL enables synchronous replication to be configured for a specific subset of the total source replicated dataset (either by inclusion or exclusion), based on application process pathname, process name, program name, or user name, thereby minimizing the impact of application latency to only the necessary transactions and data.[9]

Step Three: Monitor Application Latency

Once synchronous replication is configured and put into production, application latency needs to be monitored to ensure it stays within acceptable limits (note that all of the above-mentioned methods to manage replication latency are still applicable and available for use in a synchronous environment). Two primary configuration methods to manage (limit) application latency are:

- **SYNCLATENCYHIGH/LOW** – These parameters define values (measured in fractions of a second to seconds) for the maximum allowable measured latency[10] before Shadowbase software falls back to asynchronous replication mode (the *high* case), and the level to which the measured latency must fall before it reverts back to synchronous replication mode (the *low* case). In addition, Shadowbase software issues message alerts whenever either of these two events occur. Note that this parameter affects all synchronous transactions, i.e., when the software switches between synchronous and asynchronous modes, all synchronous transactions will be replicated either synchronously or asynchronously.

- **MAX_PREPARE_TIME** – This parameter defines the maximum length of time Shadowbase software waits before voting to commit the transaction (whether or not it has been safe-stored on the target system); it essentially limits the amount of delay contributed by Shadowbase software towards application latency.[11] This parameter is applied to individual synchronous transactions (unlike SYNCLATENCY, which applies to all transactions).

The set values of these two parameters depend upon customer-specific requirements. However, in general, the major consideration is between minimizing application latency and application response time, or ensuring that no data is lost (since triggering either the SYNCLATENCYHIGH or MAX_PREPARE_TIME parameters means that data will not be replicated synchronously and could be lost in the event of a failure).

Maximum Availability Mode (MAM)

This mode of Shadowbase ZDL processing emphasizes keeping the application running, able to process new requests through to completion, when synchronous replication cannot be assured. Shadowbase ZDL will fall back to asynchronous replication when events occur that violate the latency parameters (such as loss of connection to certain internal processes), which prevent synchronous replication from completing. Reverting to asynchronous mode when synchronous replication becomes impossible is known as Maximum Availability Mode (MAM). MAM enables application transaction processing to proceed at the possible risk of lost data if a failure occurs.

[9]A future release of Shadowbase ZDL may allow synchronous replication to be specified programmatically on a per transaction basis.
[10]What is actually measured is how far Shadowbase software is behind (lagging) in reading change data from the TMF audit trail (as the audit trail timestamp is the closest approximation of the actual time when the I/O occurred).
[11]Note that commitment processing occurs in parallel across all transaction participants (such as disk processes) – Shadowbase software is just one participant. The time taken by each participant is not cumulative with respect to the total application latency observed by the application. Generally, the observed application latency is a function of the time taken by the transaction participant that takes the longest to complete its processing, which may or may not be the Shadowbase ZDL software.

Maximum Reliability Mode (MRM)

This mode of Shadowbase ZDL processing emphasizes keeping the data safe over keeping the application running when synchronous replication cannot be assured. It is a future feature, but can be simulated with under the MAM configuration by setting the SYNCLATENCYHIGH and MAX_PREPARE_TIME parameters to infinite numbers. This mode is called Maximum Reliability Mode (MRM). In MRM mode, when synchronous replication is interrupted, Shadowbase software either aborts in-flight synchronous transactions,or holds them until the situation is resolved and synchronous replication can resume. MRM thus ensures no data could be lost during a failure, but doing so also prevents application transaction processing from proceeding until the failure is resolved.

Choosing whether to use MAM or MRM depends upon the specific requirements of each application, between the need to maintain application services or the need to not lose any application data.

Additional Latency Management Mechanisms

In addition to these mechanisms to control application latency, numerous gathered statistics may be used to monitor and drill-down into what may be causing any occurrences of excessive application latency, including:

- **COMMIT_SLA** – a user-supplied target maximum threshold time taken for both phases of the transaction commit processto complete, as seen by Shadowbase software. The number of times this value is exceeded is counted and reported (but this parameter does not limit the time taken for transaction completion to this value).

- **CMT AVG** – the average time taken for both phases of the transaction commit process to complete (the same interval measured for the COMMIT_SLA count). This value roughly approximates to the average application latency actually observed by the application.

- **PREP MIN/MAX/AVG** – the minimum, maximum, and average time taken for Shadowbase software to completethe first (prepare) phase of the transaction commit process. These values represent the amount by which Shadowbase synchronous replication processing has contributed to the observed application latency.

If an unacceptable level of application latency is detected, there are numerous tuning and configuration parameters provided by Shadowbase software which can help return it to acceptable limits. For more details of these parameters and those discussed above, please consult the *Shadowbase Synchronous Replication Manual*.

HPE NonStop TMF Parameters Affecting Application Latency

As well as the Shadowbase parameters discussed above, the HPE NonStop TMF transaction management subsystemalso provides configuration options which can affect application latency:

- **TMP Wait Timer** – the commits of all transactions in a NonStop system are coordinated by the fault-tolerant Transaction Monitor Process (TMP), which is part of TMF. The TMP can be configured to process transactions in batches to improve system performance; generally, the higher the value, the more efficient aggregated transaction processing is, however that can lead to longer application latency. The TMP Wait Timer determines how long the TMP should “sleep” before processing the next batch of transactions. It has three settings:

    1. off (process each commit immediately)
    2. auto (allow TMF to compute a wakeup interval)
    3. milliseconds (sleep this long between servicing commits that have accumulated)

    For the lowest application latency, the TMP Wait Timer should be set to “off”; this is the default setting.

- **PIO Buffer** – the PIO Buffer is used to batch broadcast messages to CPUs, informing them about transaction begins and completions. When a transaction begins, a broadcast message is sent to all CPUs in the system, advising them of this event. When the transaction completes (either as a commit or as an abort), a “prepare” broadcast message is sent to all CPUs, asking them whether they were involved in the transaction and, if so, which DP2 processes in each CPU were involved. These DP2 processes will then be asked to vote on whether the transaction should be allowed to commit, or if it should be aborted.

    When TMF decides the outcome of the transaction, another broadcast message is sent to the CPUs involved in the transaction with a list of DP2 processes in each CPU. These processes are told to commit or abort the transaction and to release their locks on the changed data.

Under heavy transaction loads, there may be a high volume of broadcast messages. To improve system performance and reduce the total number of messages sent and received, TMF can accumulate these inter-CPU broadcast messages into the PIO Buffer and send them as abatch.

There are several configuration parameters for the PIO Buffer. They include the size of the buffer, how many buffers to fill before sending, and a time in which to send the buffers if they haven't filled. The default is to send a transaction broadcast message as soon as it is received. If multiple broadcast messages have accumulated since the last broadcast, they are all sent as a batch. *This setting, the current TMF default, should be used to minimize application latency.*

IMPORTANT: it is strongly recommended not to change these default TMF settings in order to avoid unpredictable and undesirable behavior in both TMF and Shadowbase software. To minimize application latency and maximize replication performance, they should both be set to their default values.

Summary

Managing latency in NonStop systems is one of the most important aspects of configuring and operating a Shadowbase data replication environment. This technical brief reviewed control and monitoring tools that reliably limit replication and application latency, and provide timely warnings when these latencies exceed desired SLA thresholds. In the case of application latency, Shadowbase ZDL provides powerful methods to manage the trade-off between potential data loss and transaction response times,allowing the desired level of application RPO to be set – zero for data which must not be lost, or higher values when it is more important to maintain application services in the event synchronous replication cannot be assured.

Hewlett Packard Enterprise globally sells and supports Shadowbase solutions under the name ***HPE Shadowbase***. For more information, please contact your local HPE Shadowbase representative or visit our website (www.ShadowbaseSoftware.com). For additional information, please view our Shadowbase solution videos: vimeo.com/shadowbasesoftware.

Learn more:
shadowbasesoftware.com
hpe.com

Contact us:
Gravic, Inc.
17 General Warren Blvd
Malvern, PA 19355-1245 USA
Tel: +1.610.647.6250
Fax: +1.610.647.7958
Email Sales: shadowbase@gravic.com
Email Support: sbsupport@gravic.com

Please follow: