IBM Informix 14.10: Replication Performance

May 2019

Authors: Nagaraju Inturi, Vladimir Kolobrodov

Abstract

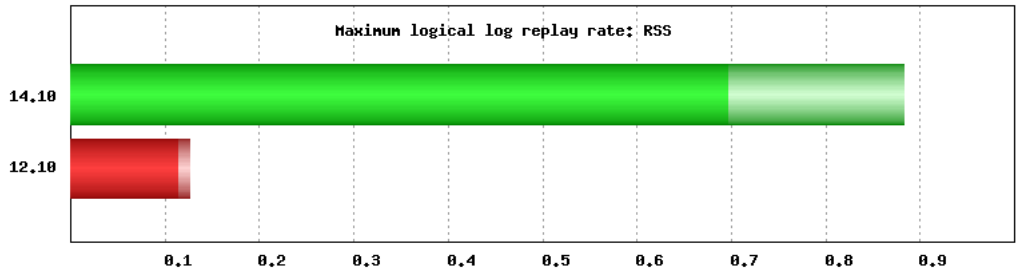
Increased processing power in current CPUs, both due to higher clock speeds and number of available cores, saw Informix deployments producing much higher transactional throughput. The replication technologies that rely on logical log replay (HDR, RSS, SDS) in 12.10 and earlier products reached a design limit, and under peak loads, a secondary instance may need significantly more time to catch up with the primary. In Informix 14.10, the logical log transfer and replay mechanism was completely overhauled, resulting in greatly improved speed. Informix now supports replication with no delay for transactional rates of up to 5 - 8 times higher compared to the prior (12.10 and earlier) Informix releases. Another functionality that greatly benefits from the performance enhancement is crash recovery - Informix 14.10.xC1 can be brought online after a crash much faster.

This document highlights the replication performance improvements in 14.10.xC1 and it can be used as a reference for optimizing the performance of an Informix cluster.

Summary

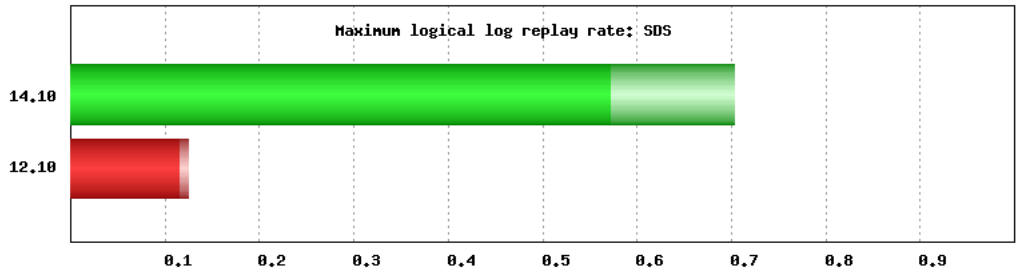
The following plots show performance improvement in the Informix 14.10.xC1 compared to the previous version, at the time of writing it is Informix 12.10.xC12, for various types of secondary servers.

Remote Standalone Secondary (RSS) - log replay performance compared to 12.10 is up to 7 times better:



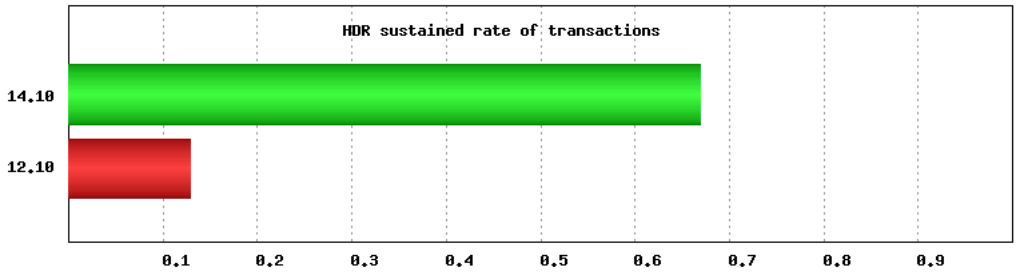
(x-axis is in millions of log records per second, higher value is better)

Shared Disk Secondary (SDS) - log replay performance compared to 12.10 is over 5 times better:



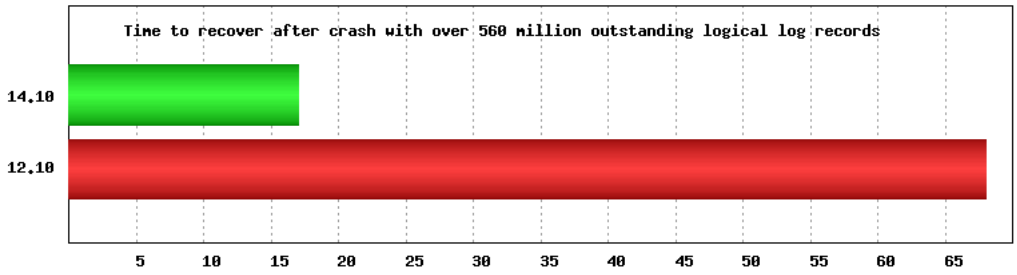
(x-axis is in millions of log records per second, higher value is better)

Sustained rate of transactions with High Availability Data Replication (HDR) setup is over 5 times better compared to 12.10



(x-axis is in millions of transactions per minute, higher value is better)

Informix 14.10 recovered from crash with over half a billion outstanding records 4 times faster compared to 12.10



(x-axis is time in minutes, lower value is better)

Details

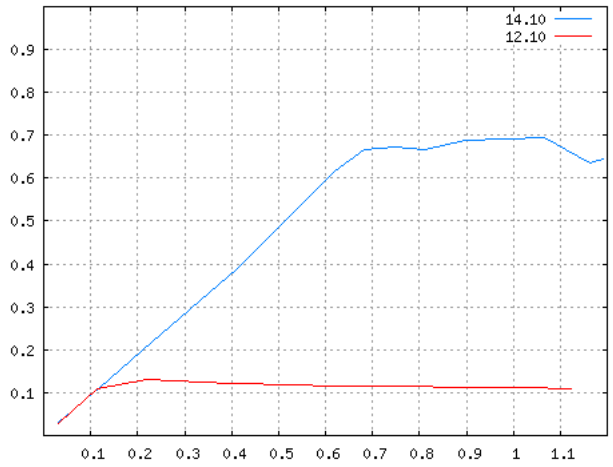
RSS log replay rate vs logical logging rate on primary

For better understanding of the replication performance improvements in Informix 14.10 we need to look at the cluster behavior under different conditions, with workloads which generate varying levels of transactional activity. Under light load the secondary instance of any type will be able to remain in sync with primary, as long as the secondary's ability to replay logical logs matches or exceeds rate with which logical log records are generated on the primary. Therefore, to investigate replication performance, an environment capable of generating high level of transactional activity is needed. The benchmark similar to industry standard TPC-C seemed like a good fit and was selected for this exercise.

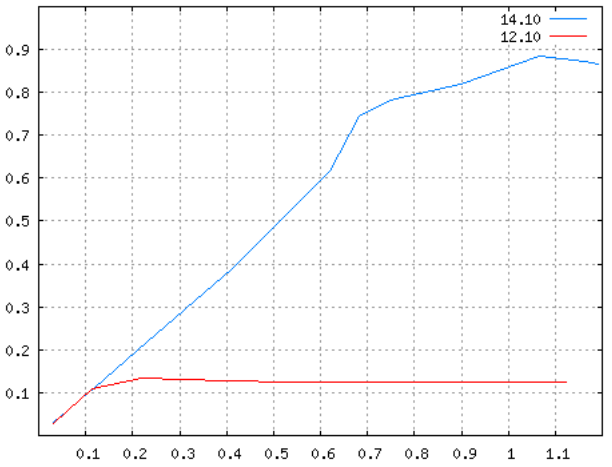
Informix setup with RSS (**Remote Standalone Secondary**) was used to execute the OLTP benchmark and measure transactional rate, the rate of logical logging on primary and the secondary log replay rate. With OLTP throughput, counting only new order transactions, reaching million transactions per minute (total transactions rate was over 50,000 per second) the logical log was written with a speed exceeding 1.1 million record per second. Several 8-minute benchmark runs were executed with varying workloads using Informix 14.10 and 12.10, in all cases waiting until RSS caught up with the primary. Then, the log replay rate was calculated for the window when benchmark was active as well as for the period when it has finished, and primary was idle aside from transferring remaining transactions (logs) to the RSS.

The plots below show the log replay rate as function of logical logging rate, both in millions of records per second. Linear region represents range where RSS can keep up with the primary with no or minimal delay.

RSS log replay rate with steady load on primary

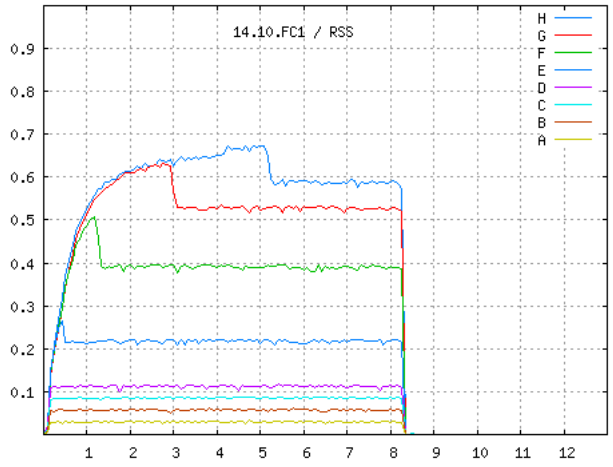


Peak RSS log replay rate with moderate or no load on primary

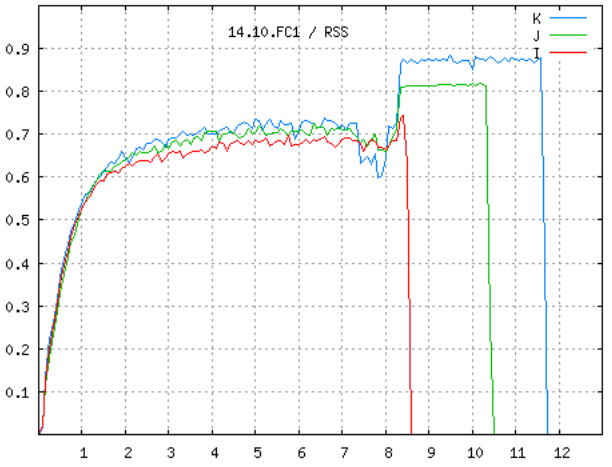


Each benchmark run lasts about 8 minutes and is executed to produce different but stable level of transactional load on the Informix primary server. This allows to determine maximum rate where RSS can still keep up, and also see what happens when logical logs are written with speed exceeding replay capability on the secondary. The plots below show log replay rate on secondary (y-axis, millions of records per second) vs time in minutes (x-axis) for a set of benchmark executions with different transactional load.

RSS can keep up or catches up



Transactional (log) rate is greater than RSS replay rate

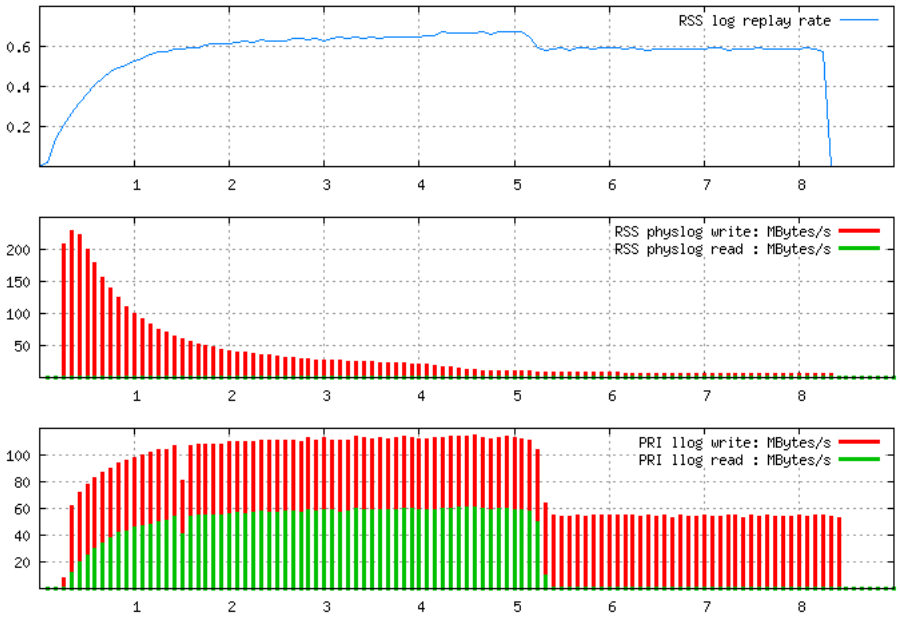


The plot containing benchmark runs "A" through "H" shows distinct pattern clearly visible for "F", "G" and "H", where log replay rate does not instantly match logical log rate generated by benchmark but is gradually increasing. That is caused by significant physical logging activity which is highest at the time benchmark starts and becomes less intense with time. When physlog activity subsides - secondary apply rate increases, exceeding logical log rate generated by the benchmark. The log replay rate continues to increase until RSS catches up with primary. Slow physical logging may limit log replay rate for the OLTP workloads. Placing Informix

RSS physical log to the device or storage subsystem with better capabilities will allow RSS to catch up with primary faster.

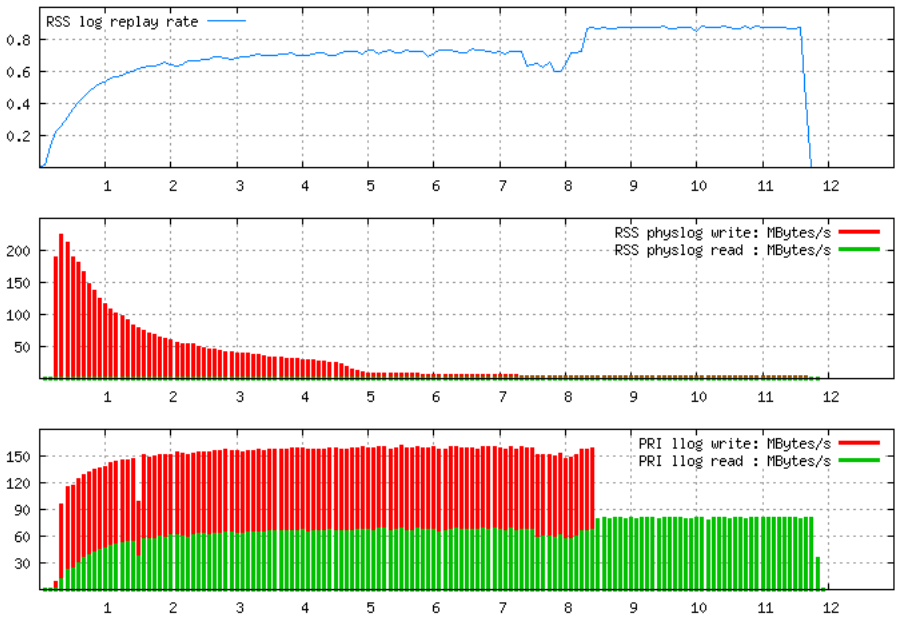
The log I/O on RSS (physical log) and Primary (logical log) for the benchmark run "H" is presented below, correlated with the log replay rate (x-axis is time in minutes):

RSS log replay rate and log I/O



The log I/O on RSS for the benchmark run "K", where benchmark runs for approximately 8 minutes and generates log records with rate exceeding 1.1 million per second on primary:

RSS log replay rate and log I/O when primary is under high transactional load



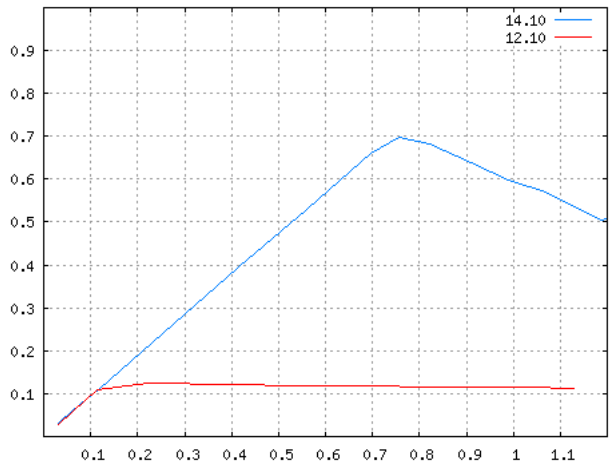
During period of high transactional load, as shown by the last plot, the logical log device on primary experiences extremely high I/O activity, read and write at the same time. It's very important to have storage subsystem that can support such I/O pattern, possibly using separate device or I/O channel for the logical log. Likewise, I/O capability of the storage device used for the RSS physical log should be appropriate for the expected peak rate of transactions. It also has to be noted that with improved log replay performance - network bandwidth and latency of the link between primary and secondary may become a limiting factor.

Shared Disk Secondary

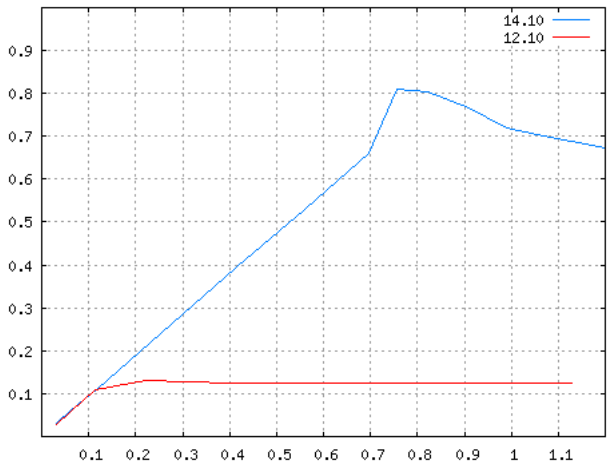
Similar data was collected for the Informix primary and SDS (Shared Disk Secondary) pair.

The plots below show comparison of the log replay rate vs logical logging rate on primary between Informix 14.10 (blue line) and 12.10 (red line). Y-Axis is the logical log replay rate, and X-Axis is the logical logging rate, both in millions of records per second. Linear region represents range where SDS keeps up with the primary with no or minimal delay.

SDS log replay rate with steady load on primary

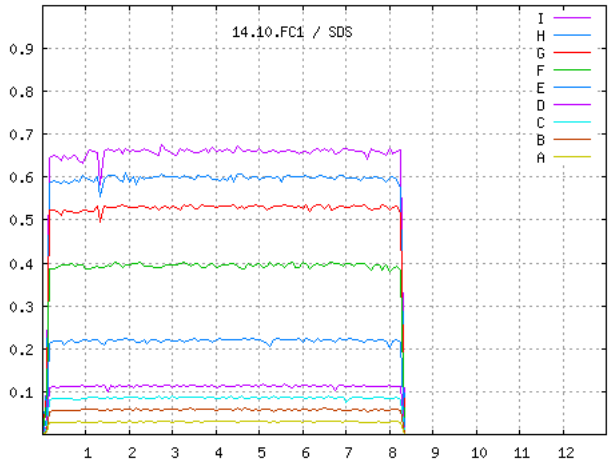


Peak SDS log replay rate with moderate or no load on primary

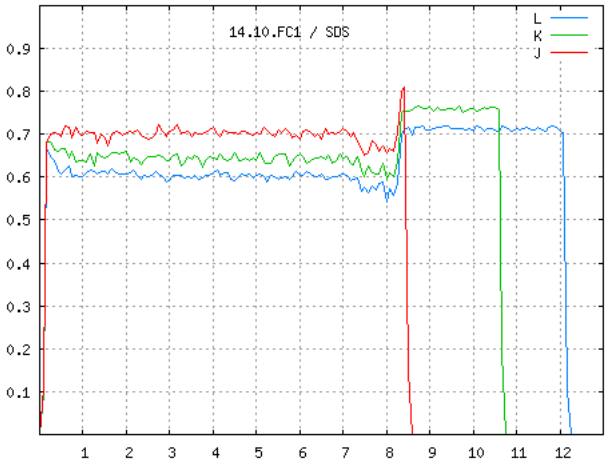


Physical logging, which affects RSS replay rate at the start of benchmark, is not an issue with SDS. The plots below, representing SDS replay rate (y-axis) vs time in minutes (x-axis) for varying levels of transactional activity, show that target apply rate - up to the maximum supported by configuration - can be reached by the SDS secondary without delay, as seen on the left plot, lines A through I. However, at or close to the peak load on primary we saw the replay rate on SDS secondary being affected, as is visible on the right plot - lines J,K,L.

SDS keeps up with primary



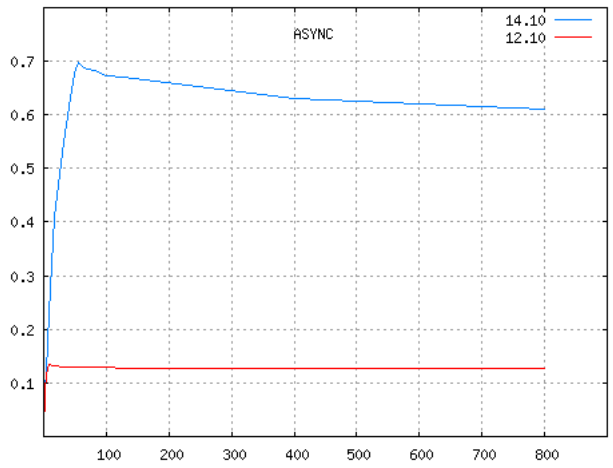
Log rate on primary exceeds SDS replay rate

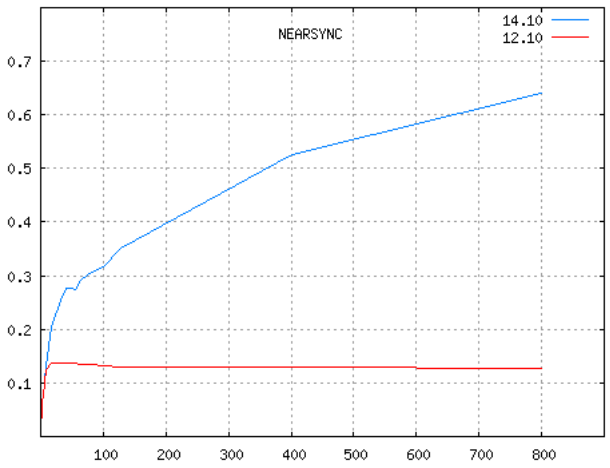


High performing storage subsystem for the logical logs remains a critical requirement for the SDS.

High Availability Data Replication

With Informix HDR the benchmark was executed for configurations where HDR_TXN_SCOPE was set to NEAR_SYNC and ASYNC (DRINTERVAL = 0) and also with DRINTERVAL = 1. In the benchmark environment best results for the Informix 14.10 were observed when HDR_TXN_SCOPE was set to ASYNC. With HDR - transactional rate on primary is limited by the secondary's apply rate, so the performance metrics is maximum achievable benchmark throughput. Coming up with a single number for the performance improvement for the HDR is a challenge, since the benchmark throughput depends on the workload specifics and maximum throughput may not be achieved in the same setup for 12.10 and 14.10. Below are plots showing throughput (y-axis, in millions of transactions per minute) vs number of active concurrent sessions (x-axis) for the HDR pair with different transaction scope (DRINTERVAL = 0):



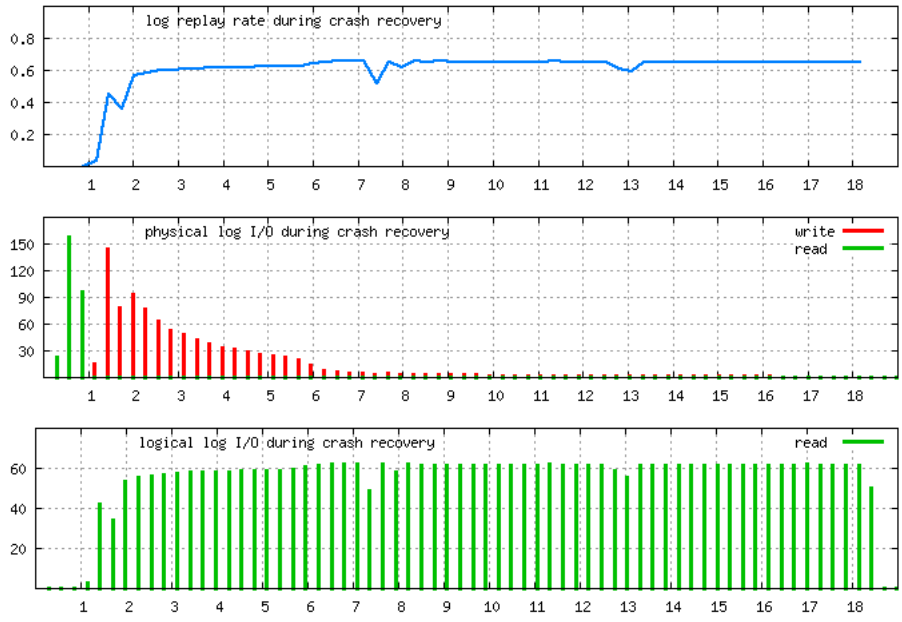


With DRINTERVAL set to "1" performance was similar to "ASYNC", but for the Informix 14.10 with the ASYNC transaction scope measured transaction latency was less compared to configuration with DRINTERVAL=1. Overall, in all tested HDR configurations Informix 14.10 demonstrated significant performance improvement compared to Informix 12.10.

Crash Recovery

Improved log replay rate also made it possible for the Informix 14.10 to recover after crash much faster compared to all previous releases.

The plot shows data collected during Informix 14.10 recovery after the instance was terminated using "kill -9" with over half a billion log records outstanding since the last checkpoint. During recovery log replay rate exceeded 0.6 million records per second. Informix 14.10 reached fully operational state over 4 times faster compared to Informix 12.10.



Configuration and Tuning

There are few configuration parameters which may be tuned for better log replay performance in the Informix 14.10. Optimal configuration depends on the hardware environment where Informix instances are deployed as well as the specifics of the workload, however, notes here may serve as a good starting point.

SMX_NUMPIPES 2

Number of SMX network connections between primary and secondary servers. Usually reasonably small number is sufficient. Recommended value is 2.

LOGBUFF 2048

The size of the logical log buffer, in KB. Also determines size of the replication buffer (see SEC_DR_BUFS). In the benchmark environment the buffered logging with LOGBUFF = 2048 was used. There is an opinion that with unbuffered logging smaller log buffer size is better, but that is not entirely true. With most production workloads having high number of active concurrent sessions using larger log buffer results in better performance. Even medium sized instance can benefit from setting this to >= 1024 regardless of the logical log mode. Smaller log buffer size with unbuffered log mode will perform better only for one or small number of concurrent sessions.

LTAPEBLK 20480

Primary server uses this configuration parameter for backing up logical logs. On the secondary 4 buffers of LTAPEBK size are used for replaying the log records.

SEC_DR_BUFS 24

Introduced in 14.10.xC1 - number of replication buffers. Applicable to RSS, SDS, HDR secondaries and primary with HDR. Buffer size is same as set by LOGBUFF configuration parameter. Minimum (default) value = 12, Maximum value = 128

SEC_LOGREC_MAXBUFS 1000

Introduced in 14.10.xC1 - number of 16k log buffers used for replaying log records. On standalone / primary instance it is in effect during crash recovery. Minimum (default) value is 4 x OFF_RECVRY_THREADS. Values over 5000 may negatively impact performance.

OFF_RECVRY_THREADS 11

The number of logical recovery threads used for log replay. On the secondary - should not exceed number of CPUVPs, and/or number of processor cores available

to the instance, especially when SEC_APPLY_POLLTIME is set to value other than "0". During crash recovery this parameter is ignored, and number of recovery threads is set to twice the number of configured CPUVPs. Recommended minimum is 5, so for secondary configured with number of CPUVPs of 5 or less the SEC_APPLY_POLLTIME parameter should probably be left at default - "0".

SEC_APPLY_POLLTIME 0

Introduced in 14.10.xC1 - time, in micro seconds, the log replay thread polls for events before yielding. May reduce thread context switch overhead while replaying log records.

Setting to value other than zero may result in increased CPU utilization during log replay and require considering changes to other Informix configuration parameters:
- poll threads may need to be moved to NETVP (NETTYPE)
- when SEC_APPLY_POLLTIME is not 0 it is recommended to set number of OFF_RECVRY_THREADS to be less than number of CPUVPs and/or number of processor cores available for the Informix instance (for the Informix instance having 64 CPUVPs and running on system with 64 logical processors with 8 threads per core set OFF_RECVRY_THREADS to less than 16)

In benchmark environment some configurations performed better with SEC_APPLY_POLLTIME set to 40, but performance across wider range of conditions was better when SEC_APPLY_POLLTIME was 0. It is recommended to tune this parameter with workload similar to production. Can be changed dynamically using "onmode -wm" or "onmode -wf".

RSS_FLOW_CONTROL
SDS_FLOW_CONTROL

For better performance flow control can be disabled by setting this parameter on **primary server** for the appropriate type of secondary to "-1". Can be changed dynamically with "onmode -wm" or "onmode -wf".

RSS_NONBLOCKING_CKPT 1

Introduced in 14.10.xC1. For the remote standalone secondary server (RSS) only - enables non-blocking checkpoint. The logs are continued to be replayed while the checkpoint is in progress. Default is "0" (log replay on RSS is blocked for the duration of the checkpoint)

DRINTERVAL 0
HDR_TXN_SCOPE ASYNC

Only applicable to HDR secondary server. In benchmark environment this resulted in overall better performance. If HDR_TXN_SCOPE must be set to NEAR_SYNC -

optimization present in Informix 14.10 may result in better performance with the **unbuffered** log mode. The HDR_TXN_SCOPE parameter is not applicable when DRINTERVAL is not 0, with DRINTERVAL = 1 performance in the benchmark was close to optimal, but with DRINTERVAL = 0 and ASYNC transaction scope - transaction delay on secondary was noticeably less.

LOG_STAGING_DIR

Must be set to a valid directory location to enable log staging at HDR while flushing buffer pool to disk during checkpoint processing. For RSS server, log staging directory configuration is a requirement to enable delay apply or stop apply. This config parameter can be set dynamically using “onmode -wf” or “onmode -wm”.

Conclusion

Considerable development effort resulted in delivery of the latest Informix release with major performance improvements in most areas of replication and high availability. With Informix 14.10 - the RSS, SDS and HDR clusters can now support much higher transactional loads with transactions propagated to secondary servers practically in real time.
At secondary server, reads and writes to logical and physical log spaces require synchronous I/O operation. For performance reasons, it is recommended to use storage supporting high number of IOPS (I/O operations per second), preferably flash based or, at least, SSD (solid state disk) medium for the log spaces.

About the authors

Nagaraju Inturi

Replication Architect at HCL Informix. Working in the field of Data Replication, Sharding, High Availability and Disaster Recovery technologies. Connect with me on LinkedIn

Vladimir Kolobrodov

Performance Architect at HCL Informix and hardware guru with a wide range of expertise running Informix in various environments, from Playstation 3 and Raspberry Pi to enterprise POWER, SPARC, and Intel systems. Connect with me on LinkedIn