

Performance of UniData Record Locking Degraded at Release 7.2.0

Abstract

Starting with UniData version 7.2.0, the number of record locks that can be set and released per second in a high-volume multi-process test is significantly lower than on previous UniData versions. Depending on your UniData application architecture, you may observe record locking performance degradation compared to your experience on prior releases.

Content

At UniData 7.2.0, an architectural change introduced the use of an operating system semaphore when a UniData process acquired or released a logical lock. While this change was made to support RESIZE CONCURRENT, the performance problem occurs even if RESIZE is never run.

The most common use of logical locks in UniData applications is for record locking via UniBasic statements like: READU, MATREADU, RECORDLOCKU, RELEASE. Any UniData lock displayed by the ECL commands LIST.READU or LIST.LOCKS is considered a logical lock.

Some performance degradation will be noticed in a single-process test. In multi-process tests - where a number of UniData processes are aggressively locking and releasing locks - the number of locks a single process can set and release in a fixed period of time is dramatically lower than at UniData 7.1. Application programs may appear to hang while making record lock calls, but will continue when they are able to acquire the new semaphore.

Depending on your UniData application architecture and processing schedules, you may or may not notice this performance problem after upgrading to version 7.2. Due to differences in how operating systems handle semaphores, UniData locking performance may vary by platform. In initial testing, HPUX and Solaris performed worse than AIX, Linux and Windows. Possible kernel tuning for semaphore performance has not yet been explored.

This has been resolved at version 7.2.4 on HPUX RISC and Itanium; 7.2.5 on AIX, Solaris, Windows, and Linux.