Database - Feature #5003

Enhance performance of H2 sequences

11/10/2020 07:08 AM - Adrian Lungu

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD
Description			

History

#1 - 11/10/2020 07:10 AM - Adrian Lungu

There is a Sequence.DEFAULT_CACHE_SIZE which defaults on 32. This refers to how often the sequence is flushed (and the session auto-commited). I feel like this default is rather low taking in consideration that now we have only one sequence for all users (if I am not mistaken). Even if there are per-user sequences (which can be triggered by using per-user physical H2 in-memory databases), it feels like this cache can be increased, so the session auto-commits more rare (1024 or 1048576 are values to be taken in account).

I am also thinking of making the sequence "temporary". Even if this is not possible in classic H2 1.4.200, there is a possibility to add a trivial patch in order to introduce temporary keyword for sequences. Recall that a temporary database object is never flushed (so no extra meta table updates and session commits).

For now, I suggest increasing the Sequence.DEFAULT_CACHE_SIZE to 1048576 as I consider to be a more reasonable value for intense used sequences. Any other optimizations at H2 sequence level will be discussed here.

#2 - 11/17/2020 06:39 AM - Adrian Lungu

The Sequence.DEFAULT_CACHE_SIZE optimization was committed in fwd-h2/rev. 4. On a singular testcase (one million nextval on a sequence), the 1048576 cache size was running in 2.45 seconds, while the default (32) cache size was running in 3 seconds. Another hot-spot in this scenario is the use of a "function based" syntax: select nextval('p2j_id_generator_sequence'). This is because the string parameter should be parsed in order to identify the schema and sequence name. Also, this parsing does not make use of PreparedStatement caching as the solution below does.

A faster approach is to use the "method based" syntax: select p2j_id_generator_sequence.nextval, which works in 1.85 seconds. This time the parser is created and used for parsing the whole statement and not the function argument alone. I suggest moving to such nextval representation.

#3 - 11/20/2020 08:32 AM - Adrian Lungu

The "method representation" has been introduced in 3821c/rev 11824 (TemporaryBuffer) and in 3821c/rev 11830 (P2JH2Dialect).

#4 - 11/07/2023 05:15 PM - Eric Faulhaber

What is the true "% Done" value of this task? It seems the improvements discussed in #5003-2 made it into trunk via 3821c. What about the ideas in #5003-1? Are those worth pursuing further?

#5 - 11/08/2023 03:51 AM - Alexandru Lungu

- Status changed from New to WIP

In the past months, while heavenly profiling FWD-H2, I removed the flushing of sequences to disk for _temp database. IMO, this is rather a flaw in the original H2: that sequences were flushed onto the disk for temporary databases, as there are used only for the lifetime of the JVM. Therefore, the cache size is no longer relevant.

The true "% Done" here is 100%, but the changes were made irrelevant by a more aggressive change in the mean-time. The cache size will still be relevant for persistent H2 (but performance tests for such scenarios weren't done that intensively). This can be closed.