Database - Bug #8451

improve performance of H2 String column comparison

03/14/2024 10:37 AM - Constantin Asofiei

Status:	Test	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#2 - 03/14/2024 10:43 AM - Constantin Asofiei

When executing the #8363 scenarios, there are ~20 million calls of CompareModeDefault.comapreString. This relies on CompareModeDefault.getKey to build a collation key.

The String table fields are actually UUIDs which are part of (unique) indexes. All these comparisons are done in a case-insensitive way.

This task is meant to find ways to improve H2's work with String indexed fields.

#5 - 03/18/2024 06:00 AM - Alexandru Lungu

The idea I have for this task:

- As most of the (string) comparisons involve 1 or even 2 database fields, I suggest caching the CollationKey inside the ValueString instance. This means we avoid using the "global" cache that hold these keys when we already store them per value instance.

Good: we avoid map look-ups and synchronizations. We can use the cached keys directly from the string instances we have. This is even better when both operands are database fields.
Slightly Bad: memory footprint will increase and performance will decrease as these will not be cross-session anymore (unless they use the common H2 value cache).
Bad: this won't fix the arbitrary strings that come as parameters. These will still use the global cache.

The caching is of course lazy (instantiate the collation key only when the string is used for the first time).

#6 - 03/18/2024 06:09 AM - Constantin Asofiei

Alexandru, the idea sounds promising - I can make the changes if you are busy with something else.

#7 - 03/18/2024 06:35 AM - Alexandru Lungu

Please go ahead. I am quite ~~stuck~~ working with some heap dumps and #8380 at the moment.

#8 - 03/19/2024 12:15 PM - Constantin Asofiei

- Assignee set to Constantin Asofiei

- % Done changed from 0 to 100

- Status changed from New to Review

Created 8451a_h2 from fwd-h2 trunk rev 41. In rev 42, I've implemented the idea in #8451-5 - cache the collation keys (case-sensitive or case-insensitive) at the ValueString instance.

Alexandru/Eric: please review. This shows a measurable improvement for #8363.

#9 - 03/20/2024 07:15 AM - Constantin Asofiei

Alexandru: can you review this today?

Greg: if review is OK, I'd like to get this merged.

#10 - 03/20/2024 07:43 AM - Alexandru Lungu

Checking out now.

#11 - 03/20/2024 07:53 AM - Alexandru Lungu

The changes are good:

- Good job in leveraging the collationKeys caches with this ValueString ci/cs keys.
- I expect to see a memory increase after this change as ValueString will retain the keys even if they evict the cache - so be it.

#12 - 03/20/2024 08:52 AM - Greg Shah

- Status changed from Review to Internal Test

Is more testing needed?

#13 - 03/21/2024 07:58 AM - Constantin Asofiei

Greg Shah wrote:

> Is more testing needed?

I'm doing ETF testing with it.

#14 - 03/21/2024 02:13 PM - Constantin Asofiei

- Status changed from Internal Test to Merge Pending

ETF testing passed, starting merge.

#15 - 03/21/2024 02:23 PM - Constantin Asofiei

Branch 8451a was created from trunk rev 15082. Rev 15083 contains the FWD-H2 build.gradle upgrade to rev 1.43.

Branch 8451a was merged into trunk as rev. 15083 and archived.

#16 - 03/21/2024 02:47 PM - Greg Shah

- Status changed from Merge Pending to Test