

Database - Feature #6895

improve performance of BufferReference methods buffer() and definition()

10/28/2022 04:35 AM - Eric Faulhaber

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		version:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to Base Language - Feature #6819: refactor FWD proxy implementation t...			New

History

#1 - 10/28/2022 04:36 AM - Eric Faulhaber

RecordBuffer.buffer() and RecordBuffer.define() (declared by the BufferReference interface, and implemented by every DMO proxy) stand out in the AspectJ method tracing as hot spots, just because they are called so often. They are invoked at the top of RecordBuffer\$Handler.invoke, if the declaring class of the method being invoked is BufferReference.

These are invoked using Utils.invoke, but considering they are used so frequently, perhaps an even more efficient, custom approach to invoking these, which bypasses the invocation handler entirely, would make sense. The ProxyAssemblerPlugin interface exists for this purpose.

If fixing the Map.get calls in Utils.invoke (see #6819) does not resolve these hot spots, we will consider this as a further action. It should be noted that doing this may be complicated, because of the "double-proxy" approach currently used to manage buffer arguments and bind special cases.

#2 - 10/28/2022 04:36 AM - Eric Faulhaber

- Related to Feature #6819: refactor FWD proxy implementation to use ReflectASM instead of Java Method reflection added

#3 - 10/28/2022 05:21 AM - Eric Faulhaber

- Project changed from Base Language to Database

#4 - 11/08/2023 04:05 AM - Alexandru Lungu

Eric / Constantin: this task is related to #6819, but it seems that it addresses a more particular scenario (.buffer(), .define()). Do you still think this task is manageable in this (November) Sprint alone (i.e. separately from #6819)? We can start working on it if it still pops out in the performance tests.