Base Language - Feature #7348

mutable proxy implementation

05/11/2023 10:47 AM - Eric Faulhaber

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

Description

Related issues:

Related to Base Language - Feature #6819: refactor FWD proxy implementation t...	**New**

History

#1 - 05/11/2023 10:52 AM - Eric Faulhaber

For certain use cases (currently only for DMO proxies), we currently proxy an existing DMO proxy to redirect method invocations from the original proxy, to its invocation handler, then to another proxy and its invocation handler.

Constantin, do you have an idea how much overhead the TemporaryBuffer proxy of a proxy solution costs, compared to just having one level of proxy? I've been thinking that instead of having a two-layer proxy implementation, we make the DMO proxy mutable. That is, the invocation handler can be swapped out, or pushed onto a stack, or something...requiring only one level of delegation/redirection, instead of two.

I don't have the details figured out, and I'm not even sure how to describe it properly yet, but it seems like it has got to be less expensive going through only one invocation handler, rather than two, for every proxied method invocation. The question is one of priority: is this enough of a hot spot in the large applications we have been profiling to warrant immediate attention? Is it used heavily enough to matter for performance?

#2 - 05/11/2023 10:53 AM - Eric Faulhaber

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

#3 - 05/11/2023 01:53 PM - Constantin Asofiei

I've looked at the profiling data and the count is this:

- ReferenceProxy.invoke - 600k via 44k instances and 12k 'bind' calls, ~35s total time
- ArgumentBuffer.invoke - 1000 times via 100 instances
- Handler.invoke - 6.7m via 60k instances, ~47s total time

For ReferenceProxy.invoke, there are 2 nested Utils.invoke calls. From these, the .buffer() call is ~50% of the calls, but the time spent is very low. But, from a total of ~35s in the profiler for ReferenceProxy.invoke, only ~50% is spent in the second Handler.invoke call.

From all Handler.invoke time, ~50% is spent in RecordBuffer.validateMaybeFlush, which is being called ~50k times.

From a cost/benefit analysis, I don't think mutable proxies will help much at this time.