Base Language - Feature #6818

analyze the performance of the FWD open client proxy client support

10/06/2022 02:37 AM - Constantin Asofiei

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	60%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		
Description			

History

#1 - 10/06/2022 02:42 AM - Constantin Asofiei

In 6129a, there are some performance improvements in DataSetSDOHelper, to cache the meta fields for a temp-table, but this 'addTableMetaData' still appears in the profiler.

A more extensive analysis is needed, to determine:

- object allocation using YourKit
- time spent in pre and post processing of the remote invocation call
- check if the metadata for dataset/tables can be avoided on transfer (if i.e. the remote side is a static temp-table, no need to transfer it, too)

#2 - 02/08/2023 02:01 PM - Constantin Asofiei

- % Done changed from 0 to 60

- Assignee set to Constantin Asofiei

- Status changed from New to WIP

I have changes force "schema marshall" to NONE in case of TABLE/DATASET parameters (although I need somehow to force this only when the parameter is being added via the FWD generated proxy client code, and not from custom/hand-written code, which may or may not match the destination).

Also, I have found a way to reduce the memory footprint of the metadata (only fields at this time) being kept in the DataGraph, by encapsulating all the info in a FieldMetaData class (this reduces the DataObject.get calls).

Next I'll spend some more time to see if I can reduce the BDT usage during transfer, especially on the temp-table serialization.

#3 - 02/21/2023 09:56 AM - Constantin Asofiei

7026a/14497 contains changes for the legacy JavaOpenClient:

- Encapsulated the field and relation metadata in their own classes, to avoid the DataObject overhead and increase performance. Other performance optimizations.
- Added an option to not marshal the table schema - this needs more analysis, as currently this flag is per-FWD connection. The API is

LegacyJavaAppserverClient.useSchemaMarshalNone().

- Refactored LegacyJavaAppserverClient APIs to always use InvokeConfig when performing remote calls.