Item 126

git_comments:

git_commits:

1. **summary:** [SPARK-31312][SQL][2.4] Cache Class instance for the UDF instance in HiveFunctionWrapper
   message: [SPARK-31312][SQL][2.4] Cache Class instance for the UDF instance in HiveFunctionWrapper ### What changes were proposed in this pull request? This patch proposes to cache Class instance for the UDF instance in HiveFunctionWrapper to fix the case where Hive simple UDF is somehow transformed (expression is copied) and evaluated later with another classloader (for the case current thread context classloader is somehow changed). In this case, Spark throws CNFE as of now. It's only occurred for Hive simple UDF, as HiveFunctionWrapper caches the UDF instance whereas it doesn't do for `UDF` type. The comment says Spark has to create instance every time for UDF, so we cannot simply do the same. This patch caches Class instance instead, and switch current thread context classloader to which loads the Class instance. This patch extends the test boundary as well. We only tested with GenericUDTF for SPARK-26560, and this patch actually requires only UDF. But to avoid regression for other types as well, this patch adds all available types (UDF, GenericUDF, AbstractGenericUDAFResolver, UDAF, GenericUDTF) into the boundary of tests. Credit to cloud-fan as he discovered the problem and proposed the solution. ### Why are the changes needed? Above section describes why it's a bug and how it's fixed. ### Does this PR introduce any user-facing change? No. ### How was this patch tested? New UTs added. Closes #28086 from HeartSaVioR/SPARK-31312-branch-2.4. Authored-by: Jungtaek Lim (HeartSaVioR) <kabhwan.opensource@gmail.com> Signed-off-by: Wenchen Fan <wenchen@databricks.com>

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-31312][SQL][2.4] Cache Class instance for the UDF instance in HiveFunctionWrapper
   body: ### What changes were proposed in this pull request? This patch proposes to cache Class instance for the UDF instance in HiveFunctionWrapper to fix the case where Hive simple UDF is somehow transformed (expression is copied) and evaluated later with another classloader (for the case current thread context classloader is somehow changed). In this case, Spark throws CNFE as of now. It's only occurred for Hive simple UDF, as HiveFunctionWrapper caches the UDF instance whereas it doesn't do for `UDF` type. The comment says Spark has to create instance every time for UDF, so we cannot simply do the same. This patch caches Class instance instead, and switch current thread context classloader to which loads the Class instance. This patch extends the test boundary as well. We only tested with GenericUDTF for SPARK-26560, and this patch actually requires only UDF. But to avoid regression for other types as well, this patch adds all available types (UDF, GenericUDF, AbstractGenericUDAFResolver, UDAF, GenericUDTF) into the boundary of tests. Credit to @cloud-fan as he discovered the problem and proposed the solution. ### Why are the changes needed? Above section describes why it's a bug and how it's fixed. ### Does this PR introduce any user-facing change? No. ### How was this patch tested? New UTs added.

github_pulls_comments:

1. **[Test build #120655 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/120655/testReport)** for PR 28086 at commit [`b515c69`](https://github.com/apache/spark/commit/b515c696799ac576bbd060a811ba2234506a2d38). * This patch **fails to build**. * This patch merges cleanly. * This patch adds no public classes.
2. To reduce the code divergence I just enabled language feature (as compiler guided on error message). There seems to be the way we don't enable it (simply use AnyRef instead of wildcard) so that's up to our preference.
3. **[Test build #120656 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/120656/testReport)** for PR 28086

at commit [`eb42ff1`] (https://github.com/apache/spark/commit/eb42ff155f220cdf8bb391160d51257bd231b222). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.

4. thanks, merging to 2.4!
5. Thanks for the quick review and merge!

github_pulls_reviews:

jira_issues:

jira_issues_comments: