

TAJO-1344

hyunsik

2015-02-12

Contents

Chapter 1

Root issue TAJO-1344

1.1 Summary

Python UDF support

1.2 Description

Python has abundant users and third-party libraries. This language is widely used in data analytic area. So, it would be great if Tajo supports Python UDF.

1.3 Attachments

1. [TAJO-1344_2.patch](#)
2. [TAJO-1344_3.patch](#)
3. [TAJO-1344_4.patch](#)
4. [TAJO-1344_5.patch](#)
5. [TAJO-1344_6.patch](#)
6. [TAJO-1344_7.patch](#)
7. [TAJO-1344_8.patch](#)
8. [TAJO-1344.patch](#)

1.4 Commits

1. Commit **a745385** by **Jihoon Son** (2015-04-18): TAJO-1344: Python UDF support. (ji-hoon)

Closes #526

1.5 Comments

1. **hyunsik**: I'm going to separate issue into udf and udaf.

2. **hyunsik:** I borrowed some of pig code. Here is the first proto type.
https://github.com/hyunsik/tajo/tree/python_udf
3. **jihoonson:** May I take this issue?
4. **hyunsik:** If so, I greatly appreciate you. Recently, I have no free time slot.
5. **hyunsik:** I just did. Thank you again.
6. **jihoonson:** This is really cool.I will happily finish your work.Thanks!
7. **jihoonson:** Hi guys, I have a suggestion for this issue.
 To support python UDFs, we need to load python codes.
 Currently, I'm only considering static loading when the tajo cluster starts up for easy development.
 To load python codes, we have two choices as follows:
 * Add some configurations to specify the paths to python codes.
 ** Example

```

1 <property>
2   <name>tajo.udf.python.path</name>
3   <value>/path/to/python/code</value>
4 </property>

```

* Support a new grammar to register python functions.
 ** Example

```

1 CREATE EXTERNAL FUNCTION foo USING python LOCATION '/path/to/python/code'

```

Definitely, the second option looks cool, but I'm worrying about the patch size.
 So, I suggest to proceed it in another Jira.

8. **yongjin.choi:** +1 for the second.
 Tajo should have a grammar to register functions after all.
 Certainly, it would be even more useful.
9. **hyunsik:** I think that we should separate the registration of program codes and the registration of function. This is because we can register more than one function from a program registration. So, we need to rethink the grammar of the second suggestion.
10. **jihoonson:** Right. The above grammar is just an example, and even wrong.
 Anywas, I'll proceed this ticket with the first suggestion.
11. **hyunsik:** I agree too. Static loading would be a good starting point.
12. **jihoonson:** Added more comments.
13. **tajoqa:** {color:red}* -1 overall.*{color} Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12707780/TAJO-1344_3.patch
 against master revision release-0.9.0-rc0-223-g373d53c.

{color:green}+1 @author.{color} The patch does not contain any @author tags.

{color:green}+1 tests included.{color} The patch appears to include 15 new or modified

test files.

{color:green}+1 javac.{color} The applied patch does not increase the total number of javac compiler warnings.

{color:green}+1 javadoc.{color} The applied patch does not increase the total number of javadoc warnings.

{color:green}+1 checkstyle.{color} The patch generated 0 code style errors.

{color:red}-1 findbugs.{color} The patch appears to introduce 3 new Findbugs (version 2.0.3) warnings.

{color:green}+1 release audit.{color} The applied patch does not increase the total number of release audit warnings.

{color:red}-1 core tests.{color} The patch failed these unit tests in tajo-catalog/tajo-catalog-common tajo-catalog/tajo-catalog-server tajo-common tajo-core tajo-plan: org.apache.tajo.engine.query.TestTrunca
org.apache.tajo.engine.query.TestCreateTable
org.apache.tajo.engine.query.TestInsertQuery
org.apache.tajo.worker.TestHistory
org.apache.tajo.engine.eval.TestSQLDateTimeTypes
org.apache.tajo.cli.tsql.commands.TestHdfsCommand
org.apache.tajo.engine.query.TestSelectQuery
org.apache.tajo.cli.tools.TestTajoDump
org.apache.tajo.ha.TestHAServiceHDFSImpl
org.apache.tajo.engine.query.TestTableSubQuery
org.apache.tajo.engine.function.TestConditionalExpressions
org.apache.tajo.engine.query.TestDropTable
org.apache.tajo.benchmark.TestTPCH
org.apache.tajo.engine.query.TestAlterTablespace
org.apache.tajo.engine.query.TestUnionQuery
org.apache.tajo.engine.query.TestSortQuery
org.apache.tajo.engine.function.TestBuiltinFunctions
org.apache.tajo.jdbc.TestResultSet
org.apache.tajo.engine.query.TestHBaseTable
org.apache.tajo.jdbc.TestTajoJdbc
org.apache.tajo.engine.planner.TestQueryValidation
org.apache.tajo.client.TestTajoClient
org.apache.tajo.engine.query.TestCaseByCases
org.apache.tajo.engine.eval.TestSQLExpression
org.apache.tajo.engine.query.TestJoinQuery
org.apache.tajo.engine.query.TestJoinBroadcast
org.apache.tajo.engine.query.TestNetTypes
org.apache.tajo.engine.query.TestSetSessionQuery
org.apache.tajo.engine.query.TestWindowQuery
org.apache.tajo.master.scheduler.TestFifoScheduler
org.apache.tajo.engine.eval.TestIntervalType
org.apache.tajo.engine.query.TestCreateDatabase
org.apache.tajo.engine.query.TestTablePartitions
org.apache.tajo.engine.eval.TestPredicates
org.apache.tajo.engine.eval.TestEvalTree
org.apache.tajo.engine.query.TestCTASQuery
org.apache.tajo.cli.tsql.commands.TestExecExternalShellCommand

org.apache.tajo.util.history.TestHistoryWriterReader
 org.apache.tajo.engine.function.TestMathFunctions
 org.apache.tajo.engine.function.TestPatternMatchingPredicates
 org.apache.tajo.cli.tsql.TestDefaultCliOutputFormatter
 org.apache.tajo.engine.function.TestDateTimeFunctions
 org.apache.tajo.engine.query.TestGroupByQuery
 org.apache.tajo.querymaster.TestQueryProgress
 org.apache.tajo.querymaster.TestTaskStatusUpdate
 org.apache.tajo.engine.function.TestPythonFunctions
 org.apache.tajo.engine.function.TestStringOperatorsAndFunctions
 org.apache.tajo.engine.query.TestNullValues
 org.apache.tajo.jdbc.TestTajoDatabaseMetaData
 org.apache.tajo.engine.planner.physical.TestSortExec
 org.apache.tajo.engine.query.TestAlterTable
 org.apache.tajo.engine.query.TestJoinOnPartitionedTables
 org.apache.tajo.storage.TestRowFile
 org.apache.tajo.engine.codegen.TestEvalCodeGenerator
 org.apache.tajo.cli.tsql.TestTajoCli
 org.apache.tajo.querymaster.TestKillQuery
 org.apache.tajo.master.TestNonForwardQueryResultSystemScanner

Test results: <https://builds.apache.org/job/PreCommit-TAJO-Build/668//testReport/>
 Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/668//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-catalog-server.html>
 Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/668//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-common.html>
 Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/668//console>

This message is automatically generated.

14. **hyunsik:** Basing on the current implementation, I suggested the first option. After TAJO-1436, you can choose the second option.

I think the following points:

- * The conventional wisdom is that a constructor should be lightweight.
- * function invocation implementation is mostly used in worker in order to evaluate expressions.
- * But, if function invocation implementation is initialized in the constructor, the initialization will be repeated several times in the following steps:
 - * in logical planner
 - * after plan deserialization in query master
 - * after plan deserialization in task runner

15. **hyunsik:** In addition, the purpose of `EvalNode::Bind(Schema)` is a preparation prior to `eval()`. Since all preparation fundamentally requires at least `Schema`, `Bind()` takes `Schema`. As you know, `eval()` took `Schema` as well as `Tuple` before TAJO-1436, and it prepares something at the first call and evaluate an expression. `Bind()` is a trial to separate the role of `eval()` into both preparation and evaluation.
16. **jihoonson:** Thanks @hyunsik. I finally agree with you. To avoid redundant work, I'd like to choose the second option. To do so, TAJO-1436 should be committed before this work. Unfortunately, you seem to not have much time. If you don't mind, may I review that issue?
17. **hyunsik:** Hi [~jihoonson],

I don't have any free slot yet. My situation would be better after one week :) Thank you for taking the review on some issues.

18. **tajoqa:** {color:red}* -1 overall.*{color} Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12724795/TAJO-1344_5.patch against master revision release-0.9.0-rc0-250-g6136142.

{color:green}+1 @author.{color} The patch does not contain any @author tags.

{color:green}+1 tests included.{color} The patch appears to include 26 new or modified test files.

{color:green}+1 javac.{color} The applied patch does not increase the total number of javac compiler warnings.

{color:green}+1 javadoc.{color} The applied patch does not increase the total number of javadoc warnings.

{color:green}+1 checkstyle.{color} The patch generated 0 code style errors.

{color:red}-1 findbugs.{color} The patch appears to introduce 31 new Findbugs (version 2.0.3) warnings.

{color:green}+1 release audit.{color} The applied patch does not increase the total number of release audit warnings.

{color:green}+1 core tests.{color} The patch passed unit tests in tajo-catalog/tajo-catalog-common tajo-catalog/tajo-catalog-server tajo-common tajo-core tajo-plan tajo-storage/tajo-storage-common tajo-storage/tajo-storage-hdfs.

Test results: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//testReport/>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-core.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-plan.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-catalog-common.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-catalog-server.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-storage-common.html>
Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/726//console>

This message is automatically generated.

19. **tajoqa:** {color:red}* -1 overall.*{color} Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12725782/TAJO-1344_6.patch against master revision release-0.9.0-rc0-255-g338a2b7.

{color:red}-1 patch.{color} The patch command could not apply the patch.

{color:red}-1 patch.{color} The patch command could not apply the patch.

Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/733//console>

This message is automatically generated.

20. **tajoqa:** {color:red}* -1 overall.*{color} Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12725784/TAJO-1344_7.patch against master revision release-0.9.0-rc0-255-g338a2b7.

{color:green}+1 @author.{color} The patch does not contain any @author tags.

{color:green}+1 tests included.{color} The patch appears to include 26 new or modified test files.

{color:green}+1 javac.{color} The applied patch does not increase the total number of javac compiler warnings.

{color:green}+1 javadoc.{color} The applied patch does not increase the total number of javadoc warnings.

{color:green}+1 checkstyle.{color} The patch generated 0 code style errors.

{color:red}-1 findbugs.{color} The patch appears to introduce 25 new Findbugs (version 2.0.3) warnings.

{color:green}+1 release audit.{color} The applied patch does not increase the total number of release audit warnings.

{color:green}+1 core tests.{color} The patch passed unit tests in tajo-catalog/tajo-catalog-common tajo-catalog/tajo-catalog-server tajo-common tajo-core tajo-plan tajo-storage/tajo-storage-common tajo-storage/tajo-storage-hdfs.

Test results: <https://builds.apache.org/job/PreCommit-TAJO-Build/734//testReport/>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/734//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-core.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/734//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-storage-common.html>
Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/734//console>

This message is automatically generated.

21. **tajoqa:** {color:red}* -1 overall.*{color} Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12725855/TAJO-1344_8.patch against master revision release-0.9.0-rc0-256-g7b78668.

{color:green}+1 @author.{color} The patch does not contain any @author tags.

{color:green}+1 tests included.{color} The patch appears to include 26 new or modified test files.

{color:green}+1 javac.{color} The applied patch does not increase the total number of javac compiler warnings.

{color:green}+1 javadoc.{color} The applied patch does not increase the total number of javadoc warnings.

{color:green}+1 checkstyle.{color} The patch generated 0 code style errors.

{color:red}-1 findbugs.{color} The patch appears to introduce 26 new Findbugs (version 2.0.3) warnings.

{color:green}+1 release audit.{color} The applied patch does not increase the total number of release audit warnings.

{color:green}+1 core tests.{color} The patch passed unit tests in tajo-catalog/tajo-catalog-common tajo-catalog/tajo-catalog-server tajo-common tajo-core tajo-plan tajo-storage/tajo-storage-common tajo-storage/tajo-storage-hdfs.

Test results: <https://builds.apache.org/job/PreCommit-TAJO-Build/736//testReport/>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/736//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-core.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/736//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-storage-common.html>
Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/736//console>

This message is automatically generated.

22. **hudson:** FAILURE: Integrated in Tajo-master-CODEGEN-build #311 (See [<https://builds.apache.org/job/Tajo-master-CODEGEN-build/311/>])
TAJO-1344: Python UDF support. (jihoon) (jihoonson: rev a7453853975dc8b5f4162ae16a1669aeb7ed995c)
* tajo-core/src/main/java/org/apache/tajo/master/TajoMaster.java
* tajo-plan/src/main/java/org/apache/tajo/plan/LogicalPlanner.java
* tajo-core/src/main/resources/python/tajo_util.py
* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/SeqScanExec.java
* tajo-plan/src/main/java/org/apache/tajo/plan/rewrite/rules/LogicalPlanEqualityTester.java
* tajo-common/src/main/java/org/apache/tajo/util/KeyValueSet.java
* tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineDeserializer.java
* tajo-common/src/main/java/org/apache/tajo/storage/Tuple.java
* tajo-core/src/test/resources/python/__init__.py
* tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/CSVLineDeserializer.java
* tajo-core/src/main/java/org/apache/tajo/worker/TajoWorker.java
* tajo-plan/src/main/java/org/apache/tajo/plan/expr/CastEval.java
* tajo-plan/src/main/java/org/apache/tajo/plan/exprrewrite/EvalTreeOptimizer.java
* tajo-core/src/test/resources/python/test_funcs2.py
* tajo-plan/src/main/java/org/apache/tajo/plan/expr/FunctionEval.java
* tajo-core/src/test/resources/queries/TestSelectQuery/testSelectWithPredicateOnPythonFunc.sql
* tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineSerDe.java
* tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/ByteBufInputChannel.java
* tajo-core/src/test/resources/python/test_funcs.pyc
* tajo-plan/src/main/proto/Plan.proto
* tajo-plan/src/main/java/org/apache/tajo/plan/rewrite/rules/PartitionedTableRewriter.java
* tajo-core/src/test/java/org/apache/tajo/engine/function/TestMathFunctions.java
* tajo-core/src/main/java/org/apache/tajo/engine/query/QueryContext.java
* tajo-plan/src/main/java/org/apache/tajo/plan/serde/EvalNodeDeserializer.java
* tajo-core/src/main/java/org/apache/tajo/engine/planner/global/rewriter/rules/GlobalPlanEqualityTester.java
* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/WindowAggExec.java
* tajo-plan/src/main/java/org/apache/tajo/plan/ExprAnnotator.java
* tajo-plan/src/main/java/org/apache/tajo/plan/expr/AlgebraicUtil.java
* tajo-core/src/test/resources/queries/TestSelectQuery/testNestedPythonFunction.sql
* tajo-core/src/main/resources/python/controller.py
* tajo-core/src/main/java/org/apache/tajo/engine/codegen/LegacyFunctionBindingEmitter.java

- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/PatternMatchPredicateEval.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/exprrewrite/EvalTreeOptimizationRule.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/OutputHandler.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/FieldSplitProcessor.java
- * tajo-core/src/main/java/org/apache/tajo/worker/TaskAttemptContext.java
- * CHANGES
- * tajo-core/src/test/java/org/apache/tajo/TajoTestingCluster.java
- * tajo-core/src/test/resources/queries/TestGroupByQuery/testGroupbyWithPythonFunc2.sql
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/function/FunctionInvocation.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/AggregationExec.java
- * tajo-catalog/tajo-catalog-common/src/main/proto/CatalogProtos.proto
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineSerializer.java
- * tajo-core/src/test/resources/results/TestSelectQuery/testNestedPythonFunction.result
- * tajo-core/src/main/java/org/apache/tajo/engine/codegen/EvalCodeGenContext.java
- * tajo-core/src/main/java/org/apache/tajo/worker/Task.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/EvalContext.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/exprrewrite/rules/ConstantFolding.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/PythonFunctionInvoke.java
- * tajo-core/src/test/resources/results/TestGroupByQuery/testGroupbyWithPythonFunc.result
- * tajo-common/src/main/java/org/apache/tajo/json/DatumAdapter.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbySecondAggregationExec.java
- * tajo-catalog/tajo-catalog-server/src/main/java/org/apache/tajo/catalog/store/MemStore.java
- * tajo-core/src/test/resources/python/test_funcs.py
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/LineSplitProcessor.java
- * tajo-core/src/test/resources/results/TestGroupByQuery/testGroupbyWithPythonFunc2.result
- * tajo-core/src/main/java/org/apache/tajo/engine/function/FunctionLoader.java
- * tajo-core/src/test/resources/queries/TestSelectQuery/testSelectPythonFuncs.sql
- * tajo-common/src/main/java/org/apache/tajo/datum/AnyDatum.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/FieldSerializerDeserializer.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/ByteBufLineReader.java
- * tajo-core/src/test/java/org/apache/tajo/engine/query/TestSelectQuery.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbyHashAggregationExec.java
- * tajo-core/src/main/java/org/apache/tajo/master/GlobalEngine.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HavingExec.java
- * tajo-storage/tajo-storage-hdfs/src/test/resources/dataset/TestJsonSerDe/testVariousType.json
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HashLeftOuterJoinExec.java
- * tajo-common/src/main/java/org/apache/tajo/datum/DatumFactory.java
- * tajo-core/src/main/java/org/apache/tajo/engine/codegen/VariablesPreBuilder.java
- * tajo-catalog/tajo-catalog-common/src/test/java/org/apache/tajo/catalog/TestFunctionDesc.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/BSTIndexScanExec.java
- * tajo-core/src/test/java/org/apache/tajo/engine/function/TestPythonFunctions.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/CSVLineSerDe.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/FunctionInvokeContext.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/python/TajoScriptEngine.java
- * tajo-common/src/main/java/org/apache/tajo/datum/BlobDatum.java
- * tajo-core/src/test/java/org/apache/tajo/engine/query/TestGroupByQuery.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/serder/EvalNodeSerializer.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/LegacyScalarFunctionInvoke.java
- * tajo-core/src/main/resources/python/__init__.py
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/CatalogUtil.java
- * tajo-docs/src/main/sphinx/functions/json_func.rst
- * tajo-common/src/main/java/org/apache/tajo/conf/TajoConf.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/InputHandler.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextFieldSerializerDeserializer.java

- * tajo-catalog/tajo-catalog-server/src/main/java/org/apache/tajo/catalog/store/AbstractDBStore.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbyThirdAggregationExec.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/EvalTreeUtil.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/Projector.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/BetweenPredicateEval.java
- * tajo-common/src/main/java/org/apache/tajo/util/datetime/DateTimeFormat.java
- * tajo-storage/tajo-storage-common/src/test/java/org/apache/tajo/storage/TestLazyTuple.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/EvalNode.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/FunctionInvoke.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/CommonJoinExec.java
- * tajo-common/src/main/java/org/apache/tajo/util/FileUtil.java
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/FunctionDesc.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/CSVLineSerializer.java
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/function/PythonInvocationDesc.java
- * tajo-storage/tajo-storage-hdfs/src/test/resources/dataset/TestDelimitedTextFile/testErrorTolerance1.json
- * tajo-catalog/tajo-catalog-server/src/test/java/org/apache/tajo/catalog/TestCatalog.java
- * tajo-core/src/test/resources/queries/TestGroupByQuery/testGroupbyWithPythonFunc.sql
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineParsingError.java
- * tajo-common/src/main/java/org/apache/tajo/QueryVars.java
- * tajo-core/src/test/java/org/apache/tajo/engine/eval/TestEvalTree.java
- * tajo-core/src/test/resources/results/TestSelectQuery/testSelectWithPredicateOnPythonFunc.result
- * tajo-plan/src/main/java/org/apache/tajo/plan/serder/LogicalNodeDeserializer.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbyFirstAggregationExec.java
- * tajo-core/src/test/java/org/apache/tajo/engine/eval/ExprTestBase.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/python/PythonScriptEngine.java
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/function/FunctionUtil.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/FieldEval.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/SelectionExec.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/StreamingUtil.java
- * tajo-core/src/test/java/org/apache/tajo/engine/eval/TestEvalTreeUtil.java
- * tajo-core/src/main/java/org/apache/tajo/master/exec/QueryExecutor.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/GeneralFunctionEval.java
- * tajo-docs/src/main/sphinx/functions.rst
- * tajo-core/src/test/resources/results/TestSelectQuery/testSelectPythonFuncs.result
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/BufferPool.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/EvalExprExec.java

23. **hudson:** FAILURE: Integrated in Tajo-master-build #673 (See [<https://builds.apache.org/job/Tajo-master-build/673/>])
- TAJO-1344: Python UDF support. (jihoon) (jihoonson: rev a7453853975dc8b5f4162ae16a1669aeb7ed995c)
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/SelectionExec.java
 - * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/BufferPool.java
 - * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/CSVLineDeserializer.java
 - * tajo-core/src/main/java/org/apache/tajo/engine/query/QueryContext.java
 - * tajo-plan/src/main/java/org/apache/tajo/plan/expr/EvalNode.java
 - * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/FunctionDesc.java
 - * tajo-plan/src/main/java/org/apache/tajo/plan/rewrite/rules/LogicalPlanEqualityTester.java
 - * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/function/PythonInvocationDesc.java
 - * tajo-core/src/test/resources/results/TestSelectQuery/testSelectPythonFuncs.result
 - * tajo-plan/src/main/java/org/apache/tajo/plan/expr/AlgebraicUtil.java
 - * tajo-plan/src/main/java/org/apache/tajo/plan/expr/FieldEval.java
 - * tajo-common/src/main/java/org/apache/tajo/datum/BlobDatum.java
 - * tajo-storage/tajo-storage-hdfs/src/test/resources/dataset/TestJsonSerDe/testVariousType.json

- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/InputHandler.java
- * tajo-common/src/main/java/org/apache/tajo/util/datetime/DateTimeFormat.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/CSVLineSerDe.java
- * tajo-common/src/main/java/org/apache/tajo/QueryVars.java
- * tajo-core/src/test/java/org/apache/tajo/engine/eval/TestEvalTreeUtil.java
- * tajo-core/src/test/resources/results/TestGroupByQuery/testGroupbyWithPythonFunc2.result
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/OutputHandler.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/CSVLineSerializer.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/global/rewriter/rules/GlobalPlanEqualityTester.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/Projector.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/LegacyScalarFunctionInvoke.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/EvalContext.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/serder/LogicalNodeDeserializer.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/EvalTreeUtil.java
- * tajo-core/src/test/java/org/apache/tajo/engine/eval/TestEvalTree.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/ByteBufInputChannel.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/FunctionInvoke.java
- * tajo-core/src/main/resources/python/tajo_util.py
- * tajo-plan/src/main/java/org/apache/tajo/plan/exprrewrite/EvalTreeOptimizationRule.java
- * tajo-core/src/main/java/org/apache/tajo/worker/TaskAttemptContext.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbyHashAggregationExec.java
- * tajo-core/src/test/java/org/apache/tajo/TajoTestingCluster.java
- * tajo-docs/src/main/sphinx/functions.rst
- * tajo-plan/src/main/proto/Plan.proto
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/EvalExprExec.java
- * tajo-core/src/test/java/org/apache/tajo/engine/function/TestPythonFunctions.java
- * tajo-core/src/test/java/org/apache/tajo/engine/function/TestMathFunctions.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/CastEval.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HavingExec.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/FunctionEval.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/python/TajoScriptEngine.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/FieldSplitProcessor.java
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/CatalogUtil.java
- * tajo-core/src/test/resources/queries/TestGroupByQuery/testGroupbyWithPythonFunc2.sql
- * tajo-core/src/main/resources/python/controller.py
- * tajo-plan/src/main/java/org/apache/tajo/plan/serder/EvalNodeSerializer.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/PythonFunctionInvoke.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/AggregationExec.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/FunctionInvokeContext.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/FieldSerializerDeserializer.java
- * tajo-common/src/main/java/org/apache/tajo/conf/TajoConf.java
- * tajo-core/src/test/resources/results/TestSelectQuery/testSelectWithPredicateOnPythonFunc.result
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/PatternMatchPredicateEval.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/rewrite/rules/PartitionedTableRewriter.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/BSTIndexScanExec.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/exprrewrite/rules/ConstantFolding.java
- * tajo-common/src/main/java/org/apache/tajo/storage/Tuple.java
- * tajo-core/src/main/java/org/apache/tajo/master/GlobalEngine.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HashLeftOuterJoinExec.java
- * tajo-catalog/tajo-catalog-common/src/main/proto/CatalogProtos.proto
- * tajo-catalog/tajo-catalog-common/src/test/java/org/apache/tajo/catalog/TestFunctionDesc.java
- * tajo-core/src/test/resources/python/__init__.py
- * tajo-core/src/main/java/org/apache/tajo/engine/function/FunctionLoader.java
- * tajo-common/src/main/java/org/apache/tajo/datum/DatumFactory.java

- * tajo-core/src/test/java/org/apache/tajo/engine/eval/ExprTestBase.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/StreamingUtil.java
- * tajo-core/src/main/java/org/apache/tajo/worker/Task.java
- * tajo-core/src/test/java/org/apache/tajo/engine/query/TestGroupByQuery.java
- * tajo-core/src/test/resources/python/test_funcs.pyc
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/ByteBufLineReader.java
- * tajo-core/src/main/java/org/apache/tajo/master/exec/QueryExecutor.java
- * tajo-core/src/main/java/org/apache/tajo/engine/codegen/EvalCodeGenContext.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextFieldSerializerDeserializer.java
- * tajo-core/src/main/java/org/apache/tajo/engine/codegen/LegacyFunctionBindingEmitter.java
- * tajo-core/src/main/java/org/apache/tajo/engine/codegen/VariablesPreBuilder.java
- * tajo-core/src/test/resources/results/TestSelectQuery/testNestedPythonFunction.result
- * tajo-common/src/main/java/org/apache/tajo/json/DatumAdapter.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/LogicalPlanner.java
- * tajo-catalog/tajo-catalog-server/src/test/java/org/apache/tajo/catalog/TestCatalog.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/python/PythonScriptEngine.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineSerializer.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/SeqScanExec.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbyFirstAggregationExec.java
- * tajo-catalog/tajo-catalog-server/src/main/java/org/apache/tajo/catalog/store/MemStore.java
- * tajo-core/src/main/java/org/apache/tajo/master/TajoMaster.java
- * tajo-common/src/main/java/org/apache/tajo/util/KeyValueType.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/exprrewrite/EvalTreeOptimizer.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/WindowAggExec.java
- * CHANGES
- * tajo-core/src/test/java/org/apache/tajo/engine/query/TestSelectQuery.java
- * tajo-storage/tajo-storage-hdfs/src/test/resources/dataset/TestDelimitedTextFile/testErrorTolerance1.json
- * tajo-plan/src/main/java/org/apache/tajo/plan/serde/EvalNodeDeserializer.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineDeserializer.java
- * tajo-catalog/tajo-catalog-server/src/main/java/org/apache/tajo/catalog/store/AbstractDBStore.java
- * tajo-storage/tajo-storage-common/src/test/java/org/apache/tajo/storage/TestLazyTuple.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/BetweenPredicateEval.java
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/function/FunctionInvocation.java
- * tajo-core/src/test/resources/python/test_funcs.py
- * tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/function/FunctionUtil.java
- * tajo-core/src/main/java/org/apache/tajo/worker/TajoWorker.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/CommonJoinExec.java
- * tajo-core/src/test/resources/queries/TestSelectQuery/testNestedPythonFunction.sql
- * tajo-core/src/test/resources/queries/TestSelectQuery/testSelectPythonFuncs.sql
- * tajo-core/src/test/resources/queries/TestGroupByQuery/testGroupbyWithPythonFunc.sql
- * tajo-common/src/main/java/org/apache/tajo/util/FileUtil.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/expr/GeneralFunctionEval.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbySecondAggregationExec.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineParsingError.java
- * tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/DistinctGroupbyThirdAggregationExec.java
- * tajo-docs/src/main/sphinx/functions/json_func.rst
- * tajo-core/src/test/resources/python/test_funcs2.py
- * tajo-core/src/test/resources/queries/TestSelectQuery/testSelectWithPredicateOnPythonFunc.sql
- * tajo-core/src/test/resources/results/TestGroupByQuery/testGroupbyWithPythonFunc.result
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/TextLineSerDe.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/ExprAnnotator.java
- * tajo-plan/src/main/java/org/apache/tajo/plan/function/stream/LineSplitProcessor.java
- * tajo-core/src/main/resources/python/__init__.py
- * tajo-common/src/main/java/org/apache/tajo/datum/AnyDatum.java

24. **jihoonson**: Committed to master.

1.6 Pull requests

1.6.1 Pull request 526

Title: TAJO-1344: Python UDF support

Author: jihoonson

Date: 2015-04-08

Status: closed

Comments:

1. **jihoonson** (2015-04-08): There are still some problems with a cluster environment. I'm investigating it.
2. **jihoonson** (2015-04-12): Ok. I think that this patch is ready for review.

To address @hyunsik's comment, I added a class called 'FunctionInvoke'. This class describes how the functions are executed.

On executing Python scripts, I used the approach of using an external UDF controller that is responsible for executing python scripts as commented above. When a submitted query involves one or more python UDFs, several UDF controllers are executed to compute UDFs. Input/output tuples are transmitted via stdio. This approach may have an issue on performance, but I think it is inevitable without using Jython.

Currently, the controller is executed for each Python functions. That is, if a query involves 5 Python functions even some of them are same, at least 5 different controllers are executed during query processing. I chose this architecture due to its simplicity.

Here are some highlights of changes.

- 'AnyDatum' is used to support Python's dynamic typing.
- 'PythonScriptEngine' is responsible for maintaining the external controller process. To reduce overhead, the controller should be forked only when UDFs are actually evaluated. In this patch, there are three points where the controller is forked.
- Constant folding optimization in Tajo master: During constant folding, some UDFs can be evaluated. If necessary, controllers are forked and immediately destroyed after evaluation.
- Non-from query execution in Tajo master. If the query involves Python UDFs, controllers are forked during query processing.
- Task execution in worker: If the plan of a stage involves Python UDFs, controllers are forked (destroyed) when a task starts up (shuts down). Due to the simplicity, I chose this architecture rather than sharing controllers among multiple tasks via 'ExecutionBlockSharedResource'.
- Refactoring the 'EvalNode::bind()' function. This function now receives 'EvalContext' in addition to 'Schema'. 'EvalContext' can contain some information given at runtime such as 'ScriptEngine' started by each task.

For reviewers, I apologize for a large patch. But many changes are related to just refactoring of the bind() function and renaming some functions.

Thanks.

3. **jinoosy** (2015-04-13): Great!! @jihoonson
I will start review. If you fix the findbug warnings in your codes, It would be better
4. **hyunsik** (2015-04-13): Thank you all guys for your efforts. I also check only some design points.
5. **jihoonson** (2015-04-16): Thanks @jinoosy and @hyunsik. I've fixed some findbugs warnings related to my patches.
6. **hyunsik** (2015-04-17): I leaved some trivial comments. The patch looks good to me. It's a great job.

Here is my additional comments. You try to change the signature EvalNode::bind to take EvalContext instance in order to get an instance of a launched script engine.

Even though this refactoring is already a breaking change. EvalNode still needs more information about task. In this chance, it would be great if we refactor EvalNode to take more meta context including TajoConf, task attempt information, and shared resources of workers. Then, we can probably remove OverridableConf parameter from all constructors of EvalNode.

If you are concerned with a large patch, we can do this work in another jira. But, it will cause the second breaking change. It is also not good as much as a large patch. You can choose either two breaking changes or a large patch. It's up to you.

7. **jihoonson** (2015-04-17): Thanks @hyunsik. I've changed the function name and removed 'QueryContextUtil' according to your comment.

Regarding on refactoring bind() function, I think that it would be better to work in another jira. For this issue, we first should decide what information are required to be passed to EvalNode. I booked another jira (<https://issues.apache.org/jira/browse/TAJO-1566>).

8. **hyunsik** (2015-04-17): I got your point and your plan. Please keep going. Here is my +1.
9. **jihoonson** (2015-04-17): I've tested on my laptop.
@jinoosy, do you have any other opinions?
10. **jinoosy** (2015-04-17): No, Looks good to me. Here is my +1
Thank you for your work

1.6.2 Pull request 473

Title: TAJO-1344: Python UDF support

Author: jihoonson

Date: 2015-03-27

Status: closed

Comments:

1. **jihoonson** (2015-03-27): I've added comments and documentations.
Please review the patch.
2. **jinossy** (2015-03-27): Great! Really useful feature
I'll review this weekend.
3. **jihoonson** (2015-03-28): I'll investigate the test failure.
4. **jihoonson** (2015-03-28): The test failure looks to be related with the version of Jython.
I fixed it and verified with 'mvn clean install -Pparallel-test,hcatalog-0.12.0 -Dmaven.fork.count=2'.
In addition, I fixed findbugs warnings.
5. **hyunsik** (2015-03-28): The patch looks good to me.

I have one suggestion. You make new EvalNode Type 'PYTHON_FUNCTION'. But, it may be not matched to the design of EvalNode, and it will require duplicated codes for EvalNode visitors.

You seem to do because Python function invocation is different from existing Java function. I think that the best solution is to separate EvalNode and Function Invocation code. In more detail, you should implement some function invocation interface, and separate java function invocation and python function invocation. Each function invocation implementation can be initialized at the first function call, or you can use EvalNode::Bind(), being addressed in TAJO-1436.

6. **jihoonson** (2015-03-29): @hyunsik, thanks for the nice suggestion. It is definitely a better solution.
However, I have a different opinion on detailed implementation.
The first option that initialize each function invocation implementation at the first function call is deprecated, so it is not a good choice.
And the second option using EvalNode::Bind() seems to initialize schema-dependent variables. Even though we still can change the specification of TAJO-1436, the current one looks good. Since relations have different schemas, schema-dependent variables should be initialized for each schema.

However, function invocation implementation does not depend on relations' schema. I think that it would be better to initialize when an instance of 'FunctionEval' is created.
What do you think?

7. **jihoonson** (2015-04-08): I'll open another PR.

Chapter 2

Connected issue TAJO-1555

2.1 Summary

Cleanup duplicated code of python functions

2.2 Description

I'm working on supporting Python UDF at TAJO-1344. This is still a prototype, and has some problems. One of the problems is related to serialization/deserialization protocol. For easy implementation, I simply used CSV format to serialize/deserialize tuples. To do so, I copied some bunch of codes from the tajo-storage package. This will incur a maintenance issue in addition to the problem of low performance.

To cleanup the duplicated codes, I think that we should use a well-known serialization/deserialization protocol such as protocol buffers.

2.3 Attachments

No attachments

2.4 Commits

No related commits

2.5 Comments

No comments

2.6 Pull requests

No pull requests