

TAJO-906

hyunsik

2014-07-04

# Contents

# Chapter 1

## Root issue TAJO-906

### 1.1 Summary

Runtime code generation for evaluating expression trees

### 1.2 Description

We have used EvalNode for two purposes:

- \* logical planning of expressions
- \* evaluation of expressions

EvalNode still is very nice for the purpose of logical planning. But, each EvalNode tree takes Datum included in a tuple and results in a Datum as a evaluation result.

So, the current approach requires many object creations, and causes interpret overheads, meaning that each one evaluation involves tree traverses and many function calls. interpretation involves also many branches, and it is harmful to CPU pipelining.

I propose Java byte code generation for each expression, and I'll use ASM (<http://asm.ow2.org/>) for it. This approach will write native java byte code, eliminating many condition branches and function calls. In addition, it is easier to deal with java primitive data types for expressions.

### 1.3 Attachments

No attachments

### 1.4 Comments

1. **hyunsik:** I'll submit the patch soon.
2. **githubbot:** GitHub user hyunsik opened a pull request:

<https://github.com/apache/tajo/pull/113>

TAJO-906: Runtime code generation for evaluating expression trees.

This is still an ongoing work. I share it for reviewing an overall approach.

Note that this patch includes some third party library because the version of ow2.asm is conflict to Hadoop's one. So, we don't need to review ow2.asm.

Also, this patch still does not pass one unit test related to IntervalDatum. I'll fix it soon.

You can merge this pull request into a Git repository by running:

```
$ git pull https://github.com/hyunsik/tajo TAJO-906
```

Alternatively you can review and apply these changes as the patch at:

```
https://github.com/apache/tajo/pull/113.patch
```

To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message:

This closes #113

```
---
commit b922a35a0f76e4ebc172d51418ed84e92837054f
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-04-02T13:18:44Z
```

added some sample codes.

```
commit 6dd5a2769d6b8be28b328647844401ddbca77c07
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-04-14T01:06:14Z
```

add some base codegen.

```
commit f735f6d35071878368f3c81591058dfe01c904a9
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-04-14T01:06:58Z
```

Merge branch 'master' of https://git-wip-us.apache.org/repos/asf/tajo into CodeGen

```
commit d9bc2f00be3b5b5d713d7c2a6691b9fffb024824
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-04-14T17:55:44Z
```

add arithmetic generation code.

```
commit 9146667dcc452156ed18ad4c46a1ea45068a9a75
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-04-15T05:11:42Z
```

Merge branch 'master' of https://git-wip-us.apache.org/repos/asf/tajo into CodeGen

```
commit a83355b12be2d6f5157156147e8634b545443aa8
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-04-16T15:02:58Z
```

Removed duplicated InvalidCastException and Refactored InvalidCastException and Invali-

dOperation.

commit 32dbcdc96c44db6c16c859877dfd855f8cb6ef3c  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-16T15:03:31Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 72fbf9c86e969f5a2c1a3dd423638d3ac816fa1d  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-17T00:38:52Z

Refactor EvalNode to be one of UnaryEval, BinaryEval, or subclass of EvalNode.

commit 1dab2aa71a1889a557542e354f810c9ad2c87908  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-17T01:38:52Z

fill visitConst()

commit ae74d2bb2d747a6aa69fe11e7d98906faa047763  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-18T15:06:08Z

Add visitField and improved ExprBaseTest to use CodeGen.

commit 8044fd9ee624c9fa3d72d64e3f106e772950825c  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-18T16:07:25Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

Conflicts:

tajo-common/src/main/java/org/apache/tajo/exception/InvalidCastException.java  
tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalTreeUtil.java  
tajo-core/src/main/java/org/apache/tajo/engine/eval/InvalidCastException.java  
tajo-core/src/main/java/org/apache/tajo/engine/planner/PlannerUtil.java  
tajo-core/src/main/java/org/apache/tajo/engine/planner/rewrite/FilterPushDownRule.java  
tajo-core/tajo-core-backend/pom.xml  
tajo-core/tajo-core-backend/src/main/java/org/apache/tajo/engine/eval/InvalidCastException.java

commit 4e1e55902e06690d2db82d91bc7c1be2fa68d758  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-19T06:08:40Z

Refactoring some method and simplified arithmetic operators.

commit af8ca5a398078f621ab7bb97b9dd187261e48469  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-19T06:09:02Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 741b1b67f833f88f4f5fd7d94140adc2fe1d350f

Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-21T19:20:24Z

add comparison code generator.

commit fe4fca74007e7b7085034d5c791883604653a221  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-22T16:51:28Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

Conflicts:

tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalTreeUtil.java  
tajo-core/src/main/java/org/apache/tajo/engine/eval/NotEval.java  
tajo-core/src/main/java/org/apache/tajo/engine/eval/SignedEval.java

commit 840eaf29f7da323a3ba88f27333cd15e56aadf33  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-22T16:51:37Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 06d08c22b75438ddb2a2bf97c5dc37c8492fcc90  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-25T20:24:58Z

passed all tests.

commit 493aa9f47e703cbb1b758503bc31606fc0fa4fd8  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-26T17:50:00Z

Improved many things.

commit a467f85b5a1bb94780782e37ab3a517285f9f1df  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-26T17:50:10Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 924984533c920a35df0d328cd1b3e86722f4180d  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-26T20:52:37Z

added unit tests and improved null handling for all operations.

commit 6255737530ef3f887f8eda22726e567cd713bafa  
Author: Hyunsik Choi <hyunsik@apache.org>  
Date: 2014-04-27T19:17:57Z

improved implicit type conversion.

commit 72bc4b06b67522e15050c29cfdc8ff3d44a5c238  
Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-04-27T19:18:14Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 54d1a7acb798a5c0134a0c249f8ba7acd23a9365

Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-04-28T14:13:11Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 61aca902f291d3184b3320828bdf36d331615fee

Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-04-28T14:55:58Z

all unit tests are passed.

commit 00a1498e99a499f9fa18bcc0c86e20c34f6680ec

Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-04-29T01:55:06Z

Moved more test from codegen.

commit a1674079f9e7cd17059e745a8dbcad511a5c23b8

Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-05-04T17:49:58Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

Conflicts:

tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/CatalogUtil.java

tajo-common/src/main/java/org/apache/tajo/datum/BooleanDatum.java

tajo-common/src/main/java/org/apache/tajo/datum/exception/InvalidCastException.java

tajo-common/src/main/java/org/apache/tajo/exception/InvalidCastException.java

tajo-common/src/main/java/org/apache/tajo/util/TUtil.java

tajo-core/src/main/java/org/apache/tajo/engine/codegen/CodeGenException.java

tajo-core/src/main/java/org/apache/tajo/engine/eval/AlgebraicUtil.java

tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalTreeUtil.java

tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalType.java

tajo-core/src/main/java/org/apache/tajo/engine/eval/SimpleEvalNodeVisitor.java

tajo-core/src/main/java/org/apache/tajo/engine/eval/UnaryEval.java

tajo-core/src/main/java/org/apache/tajo/engine/planner/ExprAnnotator.java

tajo-core/src/test/java/example/Example.java

tajo-core/src/test/java/org/apache/tajo/engine/eval/TestSQLExpression.java

commit c3dbd41cba5156eaf1badca78a4419ad78a15d18

Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-05-08T05:03:08Z

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

commit 80a9566acb8bf2ef3624d31330b46070a8dd97cf

Author: Hyunsik Choi <hyunsik@apache.org>

Date: 2014-05-08T16:41:54Z

Add between clause.

```
commit 3618d7da5a9ed1d7ea75c6f13de26d43d6ab9fd3
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-05-08T16:42:09Z
```

Merge branch 'master' of <https://git-wip-us.apache.org/repos/asf/tajo> into CodeGen

```
commit 7baff45c551920fa2c2d26d2163687d936a11bba
Author: Hyunsik Choi <hyunsik@apache.org>
Date: 2014-05-08T16:51:08Z
```

Removed the case where the length of char type is 1.

----

3. **githubbot:** Github user hyunsik commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-52966395>

I've rebased and fixed many bugs. Even though there are some trivial issues to do, this patch is ready to be reviewed.

4. **githubbot:** Github user hyunsik commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-53052194>

I've updated the patch. I've improved TajoTestingCluster to take system properties as follows:

```
““
mvn clean install -DCODEGEN=true
““
```

If “CODEGEN” is a session variable, it will be applied to QueryContext instance used in all unit tests. So, in order to test the code generation feature, you should give “-DCODEGEN=true” when you execute “mvn install”. It can be used for other session variables too.

For test for real queries, you need to set a session variable CODEGEN as follows:

```
““
tajo> \set CODEGEN true
tajo>
tajo> SELECT .....
““
```

5. **githubbot:** Github user blrunner commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-53148187>

Hi @hyunsik



This is a really great work. But I recommend for you to consider license.  
If you add ASM project license to tajo license file and tajo notice file, it would not be a problem.

Cheers

6. **githubbot:** Github user hyunsik commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-53152719>

Thanks for the review. I've updated the follows according to ASF license policy.

- \* ASM is BSD license. So, according to [Modifications to NOTICE](<http://www.apache.org/dev/licensing-howto.html#mod-notice>), I didn't modify NOTICE file.
- \* Instead, I've added ASM license to only LICENSE file.
- \* Removed ASF standard license header from ASM source codes.
- \* Please refer to [Treatment of Third-Party Works](<http://www.apache.org/legal/src-headers.html>)

7. **githubbot:** Github user blrunner commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-53153634>

I checked on ASF license deletion from ASM source codes. And I double-checked tajo LICENSE file.

'mvn clean install -DCODEGEN=true' finished successfully. And then I found that CODEGEN variable on tsql as follows:

```
1 default> \set CODEGEN true
2 default> \set
3 'CURRENT_DATABASE'='default'
4 'CODEGEN'='true'
5 'SESSION_ID'='f902f118-2649-46df-97ec-f15c2f0c75f5'
6 'USERNAME'='blrunner'
7 'SESSION_LAST_ACCESS_TIME'='1408801963706'
```

But I have a question about this patch. Is there any way of checking runtime code generation on local cluster?

8. **githubbot:** Github user hyunsik commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-53154532>

Hi @blrunner,

If CODEGEN session variable is enabled, all queries will work by using code generation. But, currently, we cannot expect its performance improvement. The code generation is designed to avoid Datum objects creations during any computations and reduce interpretation overheads like branches.

But, in order to keep the compatibility against our current Tuple and Datum mechanism, code generation feature still create lots of Datum objects. Currently, I'm working new tuple

structure using direct memory, which uses a sequence of bytes as a row blocks instead of an array of Datum objects. After than, it will give big performance benefits.

Thanks!

9. **githubbot:** Github user blrunner commented on the pull request:

<https://github.com/apache/tajo/pull/113#issuecomment-53156149>

+1

Thank you for your detailed comments.

I'm also looking forward for it to improve tajo performance. :)

10. **githubbot:** Github user asfgit closed the pull request at:

<https://github.com/apache/tajo/pull/113>

11. **hyunsik:** committed it to master branch. Thanks!

12. **hudson:** SUCCESS: Integrated in Tajo-master-build #346 (See [<https://builds.apache.org/job/Tajo-master-build/346/>])

TAJO-906: Runtime code generation for evaluating expression trees. (hyunsik: rev 7603a3d4c9ebf1e98a9f2aba53ae1)

- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/LocalVariableNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalNode.java
- \* tajo-core/src/test/java/org/apache/tajo/worker/TestFetcher.java
- \* tajo-core/src/main/java/org/apache/tajo/master/querymaster/SubQuery.java
- \* tajo-storage/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/Textifier.java
- \* CHANGES
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/CodeSizeEvaluator.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceLong.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/attrs/package.html
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/MethodOptimizer.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/CaseWhenEmitter.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/ExecutorPreCompiler.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/ByteVector.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/ProjectionExec.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/BinaryEval.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/SAXClassAdapter.java
- \* tajo-dist/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/GeneratorAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/TraceAnnotationVisitor.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/logical/HavingNode.java
- \* tajo-yarn-pullserver/src/main/java/org/apache/tajo/storage/Tuple.java
- \* tajo-common/src/main/java/org/apache/tajo/SessionVars.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Label.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Opcodes.java
- \* tajo-common/src/main/java/org/apache/tajo/datum/DatumFactory.java

- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Type.java
- \* tajo-thirdparty/asm/pom.xml
- \* tajo-core/src/main/java/org/apache/tajo/worker/TajoWorkerManagerService.java
- \* tajo-jdbc/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/ASMContentHandler.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/JSRInlinerAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/Subroutine.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/function/TestPatternMatchingPredicates.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/ClassReader.java
- \* tajo-core/src/test/java/org/apache/tajo/LocalTajoTestingUtility.java
- \* tajo-catalog/tajo-catalog-common/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/MultiANewArrayInsnNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/MethodWriter.java
- \* tajo-client/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/shrink-frames.properties
- \* tajo-common/src/main/java/org/apache/tajo/util/Pair.java
- \* tajo-core/src/main/java/org/apache/tajo/master/querymaster/QueryMaster.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/CheckAnnotationAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Context.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/AnalyzerAdapter.java
- \* tajo-catalog/tajo-catalog-client/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/ASMifier.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/IncInsnNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/FieldVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/JumpInsnNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/shrink-annotations.properties
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/SourceInterpreter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/CheckFieldAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/SerialVersionUIDAdder.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/shrink-signatures.properties
- \* tajo-catalog/tajo-catalog-server/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/shrink-writer.properties
- \* tajo-core/src/main/java/org/apache/tajo/master/LaunchTaskRunnersEvent.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/ExternalSortExec.java
- \* tajo-dist/src/main/bin/tajo
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Item.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/ConstantPool.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/signature/SignatureVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/package.html
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceString.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/SeqScanExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/ClassOptimizer.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/InstructionAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/SAXAnnotationAdapter.java
- \* tajo-rpc/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/CheckMethodAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/Frame.java
- \* tajo-docs/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/AnnotationConstantsCollector.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/NLeftOuterJoinExec.java

- \* tajo-core/src/main/java/org/apache/tajo/worker/ExecutionBlockSharedResource.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/eval/TestSQLExpression.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/shrink.properties
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalTreeUtil.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceTimestamp.java
- \* tajo-core/src/main/java/org/apache/tajo/master/TajoMaster.java
- \* tajo-core/src/main/java/org/apache/tajo/worker/TajoWorker.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/BasicPhysicalExecutorVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/FrameNode.java
- \* tajo-core/src/main/proto/TajoWorkerProtocol.proto
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/TryCatchBlockSorter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/BasicValue.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/MergeJoinExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/InsnNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/ClassConstantsCollector.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/VariablesPreBuilder.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/EvalCodeGenContext.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/Value.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/Printer.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/query/QueryUnitRequestImpl.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/EvalCodeGenerator.java
- \* tajo-storage/src/main/java/org/apache/tajo/storage/FrameTuple.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/InnerClassNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/RemappingMethodAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/TableSwitchGenerator.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/MethodConstantsCollector.java
- \* tajo-core/src/test/java/org/apache/tajo/TajoTestingCluster.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/Analyzer.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/InsnList.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/AnalyzerException.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/JarOptimizer.java
- \* tajo-core/src/main/java/org/apache/tajo/master/TajoContainerProxy.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HashLeftSemiJoinExec.java
- \* tajo-maven-plugins/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/package.html
- \* tajo-core/src/main/java/org/apache/tajo/worker/TajoResourceAllocator.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Edge.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/SelectionExec.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/PhysicalPlannerImpl.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/Shrinker.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/CheckSignatureAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/LookupSwitchInsnNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/LikePredicateEval.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/Projector.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/BSTIndexScanExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/TraceFieldVisitor.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceDouble.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/planner/physical/TestPhysicalPlanner.java
- \* tajo-core/src/main/java/org/apache/tajo/worker/Task.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/RightOuterMergeJoinExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/TryCatchBlockNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HashJoinExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/ClassWriter.java
- \* tajo-algebra/pom.xml

- \* tajo-catalog/tajo-catalog-drivers/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/jdk1.3.1\_19.txt.gz
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/logical/ScanNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/logical/SelectionNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/PhysicalPlanUtil.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/TajoClassLoader.java
- \* tajo-storage/src/test/java/org/apache/tajo/storage/parquet/TestReadWrite.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/PhysicalPlanningException.java
- \* tajo-storage/src/main/java/org/apache/tajo/storage/VTuple.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/TraceSignatureVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/AdviceAdapter.java
- \* tajo-storage/src/main/java/org/apache/tajo/storage/Tuple.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/BNLJoinExec.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/eval/TestPredicates.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/RemappingClassAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/ASMifiable.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/SAXAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/package.html
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/FieldConstantsCollector.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/eval/ExprTestBase.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/package.html
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/TraceClassVisitor.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/SimilarToPredicateEval.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/PhysicalExec.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/codegen/TestEvalCodeGenerator.java
- \* pom.xml
- \* tajo-catalog/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/TraceMethodVisitor.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/PartitionMergeScanExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/BasicVerifier.java
- \* tajo-yarn-pullserver/pom.xml
- \* tajo-jdbc/src/main/java/org/apache/tajo/jdbc/MetaDataTuple.java
- \* tajo-core/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/SmallSet.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/BasicInterpreter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/LabelNode.java
- \* tajo-core/src/test/resources/results/TestTajoCli/testHelpSessionVars.result
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/PatternMatchPredicateEval.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/CheckClassAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/ClassVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/Textifiable.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/RemappingFieldAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/FieldWriter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/SimpleRemapper.java
- \* tajo-project/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/StaticInitMerger.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/SAXCodeAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/Remapper.java
- \* tajo-core/src/main/java/org/apache/tajo/master/GlobalEngine.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/ExprAnnotator.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Handle.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/signature/SignatureReader.java
- \* tajo-common/src/main/java/org/apache/tajo/conf/TajoConf.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/CaseWhenSwitchGenerator.java

- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Frame.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/MethodInsnNode.java
- \* tajo-common/src/main/java/org/apache/tajo/datum/CharDatum.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceTime.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/MethodVisitor.java
- \* tajo-common/src/main/java/org/apache/tajo/datum/Datum.java
- \* tajo-common/pom.xml
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HashFullOuterJoinExec.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/EvalType.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/Processor.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/CodeGenUtils.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/TableSwitchInsnNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/Constant.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/RemappingSignatureAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/InvokeDynamicInsnNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/UnaryPhysicalExec.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/TajoGeneratorAdapter.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceBoolean.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/RemappingAnnotationAdapter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Attribute.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/SimpleVerifier.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/LocalVariablesSorter.java
- \* tajo-catalog/tajo-catalog-drivers/tajo-hcatalog/pom.xml
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/package.html
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/SourceValue.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/util/package.html
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/shrink-resize.properties
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/signature/package.html
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/AnnotationNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/CompilationError.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/codegen/TestGeneratorAdapter.java
- \* LICENSE
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/AnnotationWriter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/ClassNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/signature/SignatureWriter.java
- \* tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/FunctionDesc.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/HashLeftOuterJoinExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/Handler.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/codegen/EvalCodeEmitter.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/PhysicalExecutorVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/FieldNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/MethodNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/jdk1.2.2\_017.txt.gz
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/LdcInsnNode.java
- \* tajo-storage/src/main/java/org/apache/tajo/storage/parquet/ParquetAppender.java
- \* tajo-common/src/main/java/org/apache/tajo/util/CommonTestingUtil.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/TypeInsnNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/function/builtin/CoalesceDate.java
- \* tajo-storage/src/main/java/org/apache/tajo/storage/LazyTuple.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/AbstractInsnNode.java
- \* tajo-core/src/main/java/org/apache/tajo/worker/TaskAttemptContext.java
- \* tajo-core/src/test/java/org/apache/tajo/engine/function/TestConditionalExpressions.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/NLJoinExec.java

- \* tajo-catalog/tajo-catalog-common/src/main/java/org/apache/tajo/catalog/CatalogUtil.java
- \* tajo-common/src/main/java/org/apache/tajo/util/datetime/DateTimeUtil.java
- \* tajo-common/src/main/java/org/apache/tajo/datum/Inet4Datum.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/FunctionEval.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/commons/Method.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/LineNumberNode.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/asm-xml.dtd
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/AnnotationVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/analysis/Interpreter.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/IntInsnNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/eval/SimpleEvalNodeVisitor.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/optimizer/NameMapping.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/xml/SAXFieldAdapter.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/MergeFullOuterJoinExec.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/VarInsnNode.java
- \* tajo-common/src/main/java/org/apache/tajo/datum/NullDatum.java
- \* tajo-thirdparty/asm/src/main/java/org/apache/tajo/org/objectweb/asm/tree/FieldInsnNode.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/physical/BinaryPhysicalExec.java
- \* tajo-core/src/main/java/org/apache/tajo/engine/planner/logical/SelectableNode.java

## Chapter 2

# Connected issue TAJO-1041

### 2.1 Summary

Tuple Block Iteration Model

### 2.2 Description

Currently, our physical executor is based on a tuple-at-a-time model. It is a traditional model and is well known that it is not efficient in terms of CPU costs.

Also, I think that we can design integrated interface for row block with considering row block and vectorized row block. I'll add some possibility to extend row block iteration model to be vectorized.

In addition, in order to achieve more CPU efficient implementations, we will make good use of off heap data structure (TAJO-907) and code generation (TAJO-906).

This is an umbrella issue, which is to change current execution model to block iteration model.

### 2.3 Attachments

No attachments

### 2.4 Comments

1. **hyunsik:** I'm designing push model iteration model instead of traditional pull-based model. Its advantages is to reduce memory copy and reuse allocated row block.

The description is as follows:

h3. RowBlock allocations

- \* Each physical executor will have its in/out row blocks, which are allocated once and are reused during a task.

- \* Each physical executor which bypasses tuples will have an output row block.

- \* Each physical executor like join, projection, and scan, which builds new tuple will have input row block and output row block.

- \*\* We call this kind of physical executor \*projectable executor\*.

- \*\* An output row block of one physical executor is the input row block of its parent physical



executor.

- \* Each physical executor which plays a role of filter will not have any row block.

- \*\* We call this kind of physical executor \*selectable executor\*. Having and Selection are this type.

- \* RowBlock interface will have selection IDs, which indicate rows matched to filter conditions of descendant nodes.

h3. Iteration model

h4. Projectable executors

- \* A projectable executor will call its child node (or nodes) with its input row block.

- \* Then, a projectable executor will fill its output row block with processed rows.

h4. Selectable executors

- \* A selectable executor will use only its output row block, which actually is an input row block of its parent row block.

- \* It will add indices of qualified rows of its filter condition to the row block.

## Chapter 3

# Connected issue TAJO-900

### 3.1 Summary

Reducing memory usage during query processing

### 3.2 Description

Currently, we have used tuple structures implemented as Java objects. It internally uses Datum objects. Current Tuple structure occupies in JVM heap space. As a result, it is hard to control memory usage, and it is impossible to predict garbage collection. This problem usually becomes severe when Tajo deals with very large data in relatively small cluster and lots of grouping or join keys.

I've tried various tests and I made some prototype to show the possibility to eliminate this problem.

The main idea is as follows:

- \* Do not use Datum class in expression evaluation. Instead, we should use java primitive type values
- \*\* It will significantly reduces object creations and memory usages
- \* Redesign Tuple using direct memory allocation (DirectByteBuffer or Unsafe.allocateMemory)
- \*\* It allows each worker to control memory usages during in-memory operations like sort and hash aggregation/joins.
- \*\* It enables column values to be stored in adjacent memory, improving cache locality.

In order to achieve the above idea, we should do as follows:

- \* implement an alternative (i.e., runtime byte code generation) to EvalNode framework in order to avoid use of Datum and Java objects.
- \* Design new tuple data structure using direct memory allocation
- \* Refactor existing operators to be controlled according current memory usage

This is an umbrella issue. I'll create subtasks, and I've already started some issues. I'll use this jira to track them.

### 3.3 Attachments

No attachments

### 3.4 Comments

1. **hyunsik:** The main purpose of this issue was achieved by TAJO-906 and TAJO-907. The remain works will be addressed in TAJO-1041.