Сторонние движки исполнения для Apache Spark: опыт использования

Никита Благодарный
telegram @nblagodarnyy

ЧЕСТНЫЙ ЗНАК | центр развития перспективных технологий

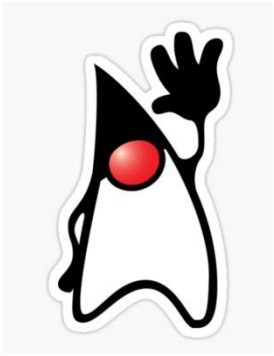
SmartData
2025

Немного вводных

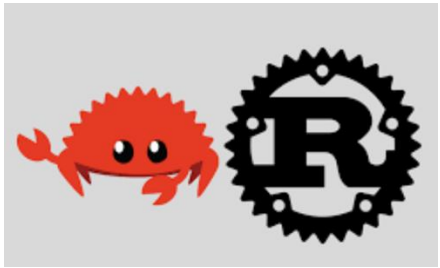
Про что этот доклад?

Изменение среды исполнения Spark

```
== Physical Plan ==
AdaptiveSparkPlan (69)
+- == Final Plan ==
   Execute InsertIntoHadoopFsRelationCommand (44)
   +- WriteFiles (43)
      +- ShuffleQueryStage (42), Statistics(sizeInBytes=90.7 GiB, rowCount=1.45E+8)
         +- Exchange (41)
            +- * Project (40)
               +- * SortMergeJoin LeftOuter (39)
                  :- * Sort (16)
                  :  +- ShuffleQueryStage (15), Statistics(sizeInBytes=63.5 GiB, rowCount=1.45E+8)
```



```
== Physical Plan ==
AdaptiveSparkPlan (69)
+- == Final Plan ==
   Execute InsertIntoHadoopFsRelationCommand (44)
   +- WriteFiles (43)
      +- ShuffleQueryStage (42), Statistics(sizeInBytes=90.7 GiB, rowCount=1.45E+8)
         +- Exchange (41)
            +- * Project (40)
               +- * SortMergeJoin LeftOuter (39)
                  :- * Sort (16)
                  :  +- ShuffleQueryStage (15), Statistics(sizeInBytes=63.5 GiB, rowCount=1.45E+8)
```



О себе

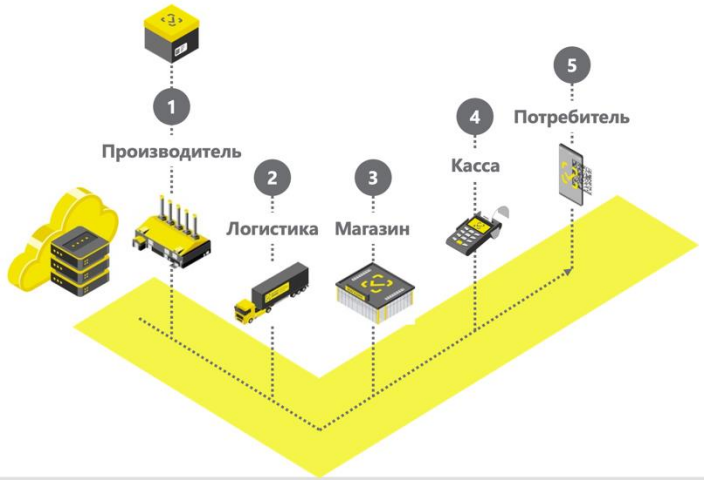
Data Engineer, 18 лет в IT, из них 15 в области BI/DE

- 2007-2011 Сбербанк, разработка OLTP-решений на Oracle, ХД/Reporting на стеке Microsoft
- 2012-2016 Сбербанк, разработка ЦХД на Oracle / Teradata / Informatica PC
- 2016-2018 КРОК, проекты на Oracle / Informatica / Hadoop (HDP, Arenadata) / Spark / Hive / NiFi
- 2018-2019 Газпромнефть, построение DataLake (Arenadata, MS SQL, Informatica, Spark/Hive)
- 2019-.... Честный знак, DataLake (Vanilla Hadoop/ Spark / Hive / Hbase / ClickHouse / Postgres etc.)

О компании Честный знак

Создаем систему цифровой маркировки и прослеживания товаров "Честный Знак" в России и странах ЕАЭС

- Основана в 2017 году
- > 1000 сотрудников
- Офисы разработки в Москве, Питере, Пензе, Ульяновске, Нижнем Новгороде
- Разнообразные товарные группы (Табак, Молоко, Вода, Пиво, Лекарства, Одежда, Обувь, ...)



SmartData 2025

О технологическом ландшафте

4 кластера (~ 600 машин)
~10 Pb хранения

Кластер 590 серверов
202 Tb RAM
15 Pb хранения

Кластер 86 машин
800 Tb хранения

2 кластера Patroni
~2 Tb хранения

Кластер 18 машин
Ёмкость – 1,1 Pb

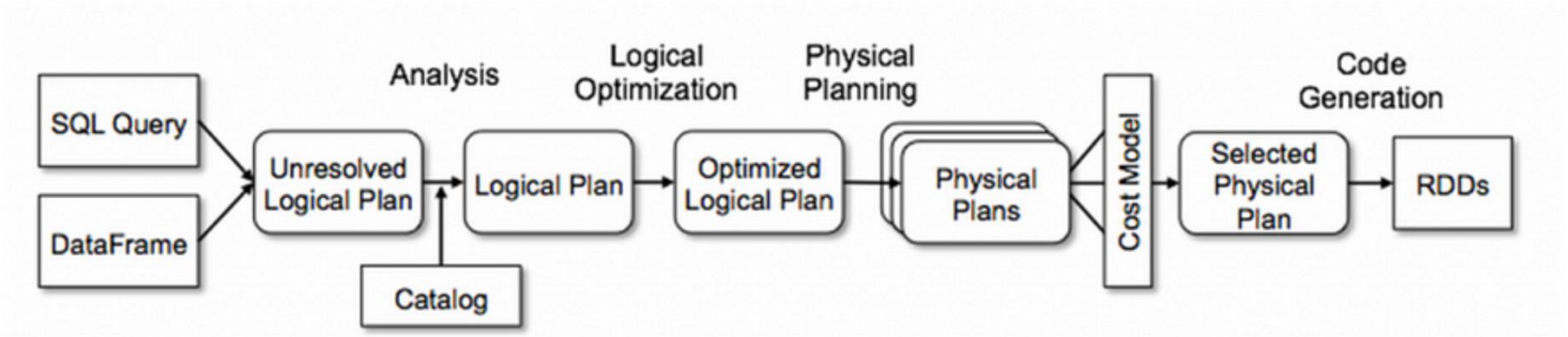
kubernetes

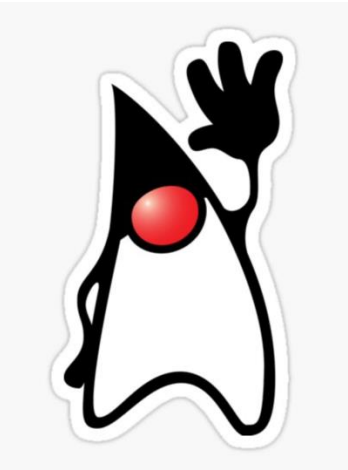
Содержание

Как работает Spark?

Catalyst

Catalyst Optimizer



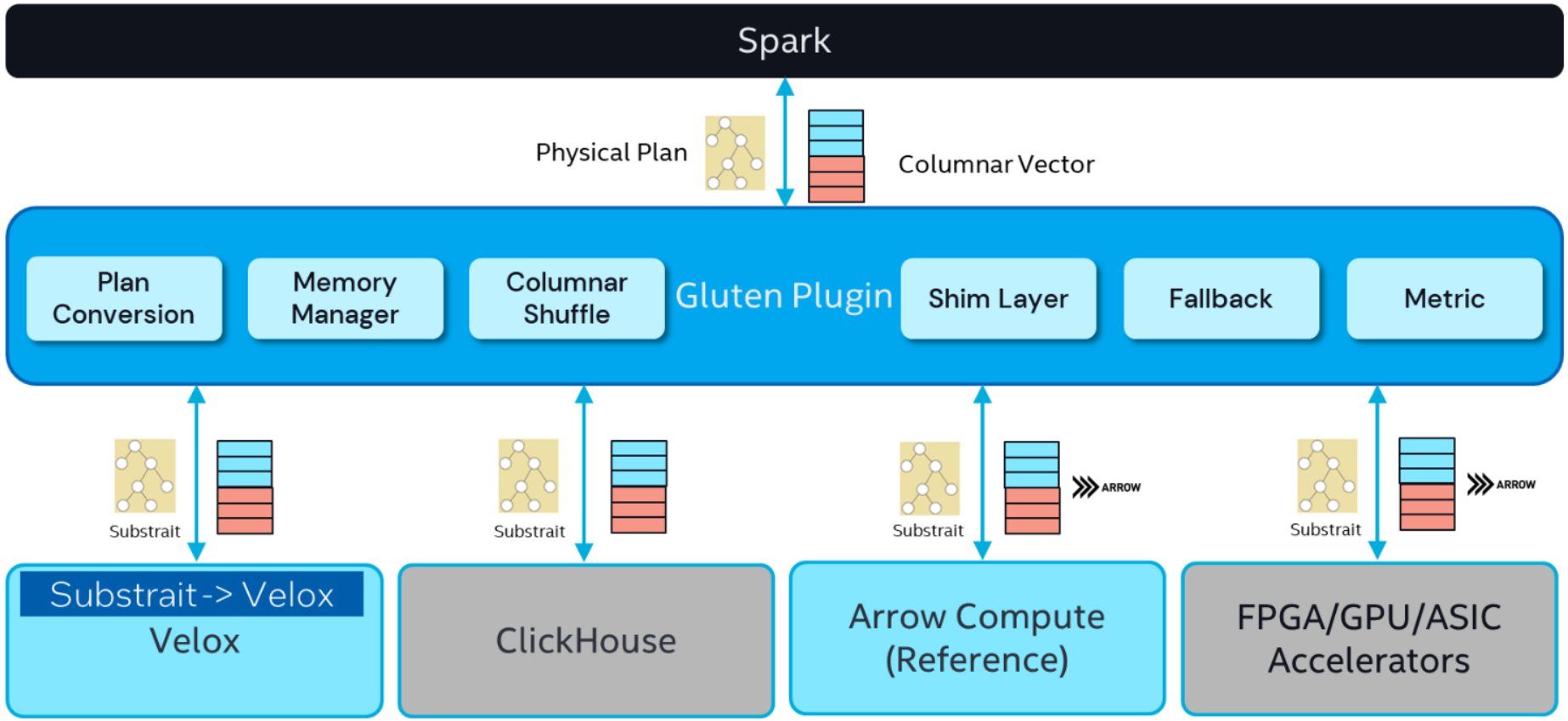


Статья Андрея Кучерова на habr.com

Доклад Димы Вертлиба на 43tech meetup

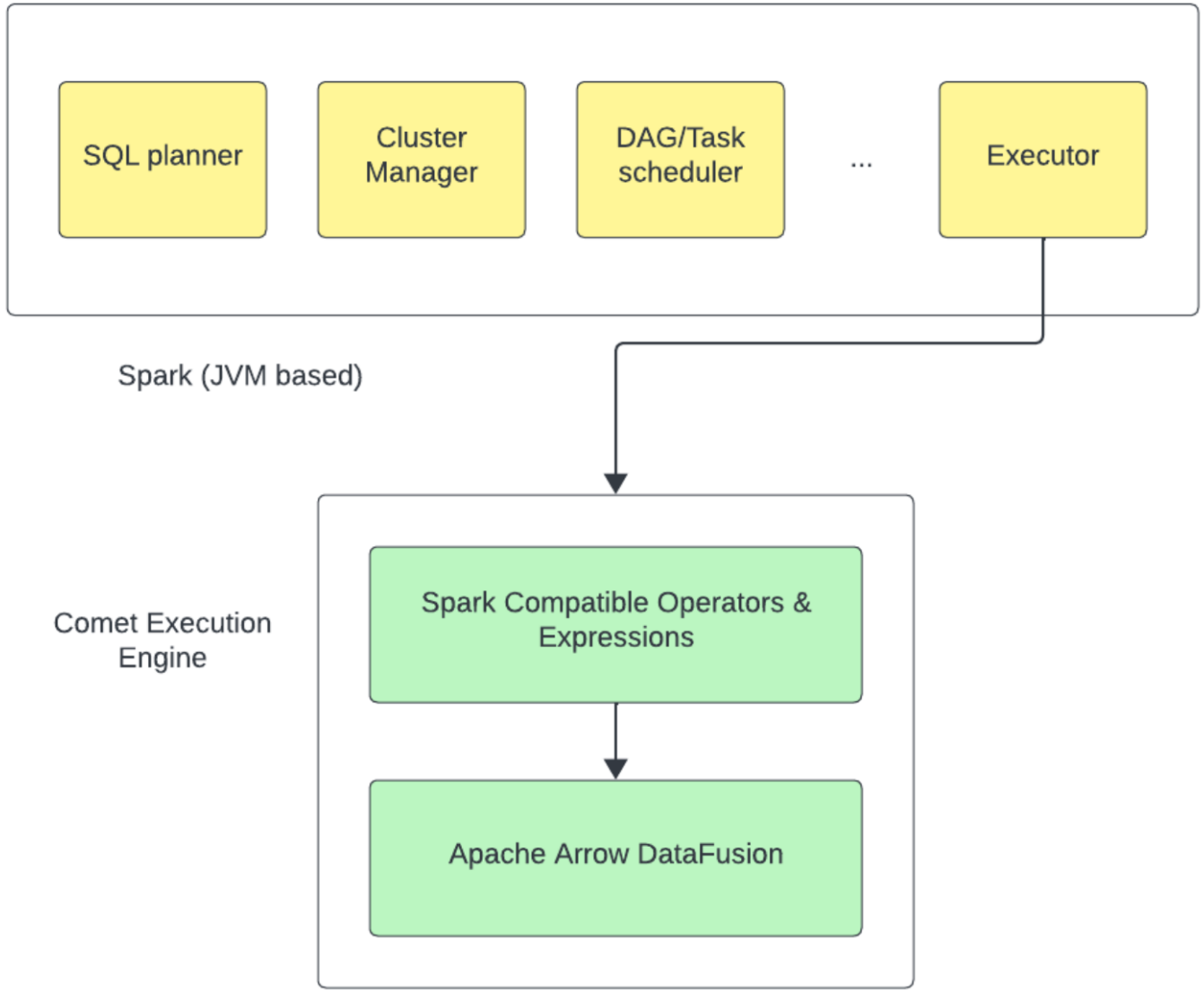
Как работают движки?

Gluten/Velox



Про модульные БД - доклад Павла Солодовникова на SmartData 2024

Comet


SQL planner
Cluster Manager
DAG/Task scheduler
...
Executor
Spark (JVM based)
Comet Execution Engine
Spark Compatible Operators & Expressions
Apache Arrow DataFusion

Охват Gluten

incubator-gluten

- Home
- Getting-Started
- **Velox Backend's Supported Operators & Functions**
- Velox Backend Limitations
- Developer Overview
- Velox Backend Troubleshooting
- Gluten Release
- Aggregate Functions Support Status
- Generator Functions Support Status
- Scalar Functions Support Status
- Window Functions Support Status

Operator Map

Gluten supports 30+ operators (Drag to right to see all data types)

Executor	Description	Gluten Name
FileSourceScanExec	Reading data from files, often from Hive tables	FileSourceScanExecTransformer
BatchScanExec	The backend for most file input	BatchScanExecTransformer
FilterExec	The backend for most filter statements	FilterExecTransformer
ProjectExec	The backend for most select, withColumn and dropColumn statements	ProjectExecTransformer
HashAggregateExec	The backend for hash based aggregations	HashAggregateBaseTransformer
BroadcastHashJoinExec	Implementation of join using broadcast data	BroadcastHashJoinExecTransformer
ShuffledHashJoinExec	Implementation of join using hashed shuffled data	ShuffleHashJoinExecTransformer
SortExec	The backend for the sort operator	SortExecTransformer

Scalar Functions Support Status

Out of 357 scalar functions in Spark 3.5, Gluten currently fully supports 239 functions and partially supports 24 functions.

Array Functions

Aggregate Functions Support Status

Out of 62 aggregate functions in Spark 3.5, Gluten currently fully supports 54 functions and partially supports 1 function.

Aggregate Functions

Window Functions Support Status

Out of 9 window functions in Spark 3.5, Gluten currently fully supports 9 functions.

Window Functions

Generator Functions Support Status

Out of 7 generator functions in Spark 3.5, Gluten currently fully supports 7 functions.

Generator Functions

Охват Comet

Search the docs ...

OVERVIEW:

- Comet Overview
- Comparison with Gluten

USER GUIDES:

- Comet 0.10.0-SNAPSHOT
- Comet 0.9.x
  - Installing Comet
  - Building From Source
  - Supported Data Sources
  - Supported Data Types
  - **Supported Operators**
  - Supported Expressions
  - Configuration Settings
  - Compatibility Guide
  - Tuning Guide
  - Metrics Guide
  - Iceberg Guide
  - Kubernetes Guide
- Comet 0.8.x

CONTRIBUTOR GUIDE

- Getting Started

Supported Spark Operators

The following Spark operators are currently replaced with native versions. Query stages that contain any operators not supported by Comet will fall back to regular Spark execution.

Operator	Notes
Projection	
Filter	
Sort	
Hash Aggregate	
Limit	
Sort-merge Join	
Hash Join	
BroadcastHashJoinExec	
Shuffle	
Expand	
Union	

Подсчёт общего количества у всех категорий:

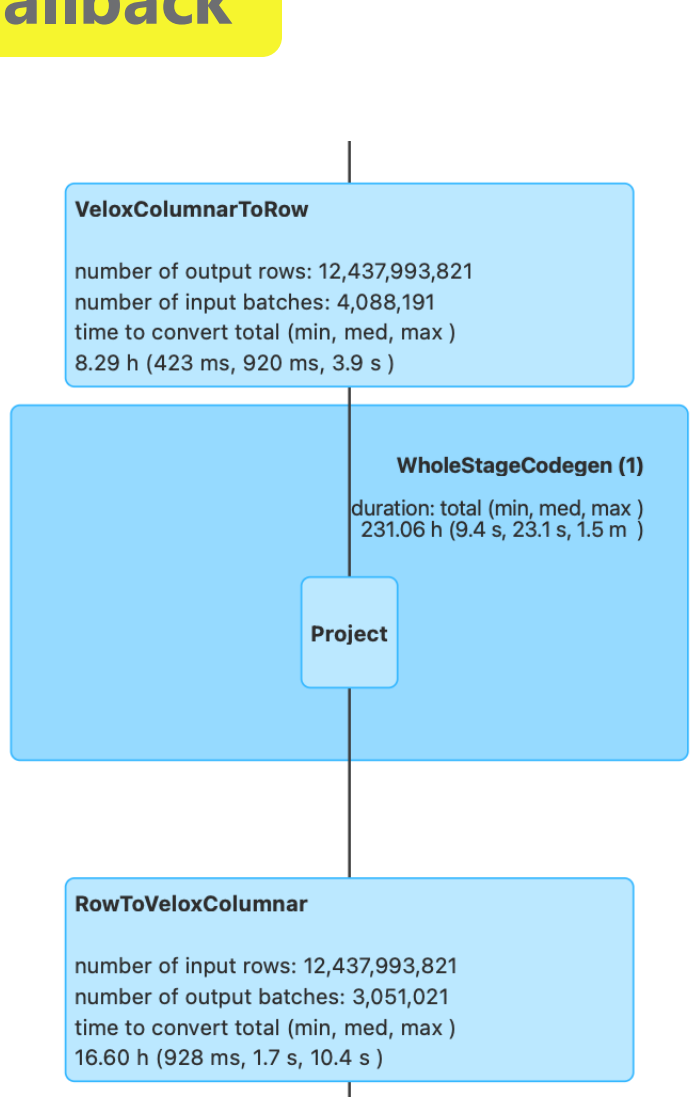
Категория	Количество выражений
Literal Values	1
Unary Arithmetic	1
Binary Arithmetic	6
Conditional Expressions	2
Comparison	9
String Functions	25
Date/Time Functions	8
Math Expressions	19
Hashing Functions	4
Boolean Expressions	3
Bitwise Expressions	8
Aggregate Expressions	17
Arrays (experimental)	15
Structs	3
Other	6
Итого	**128**

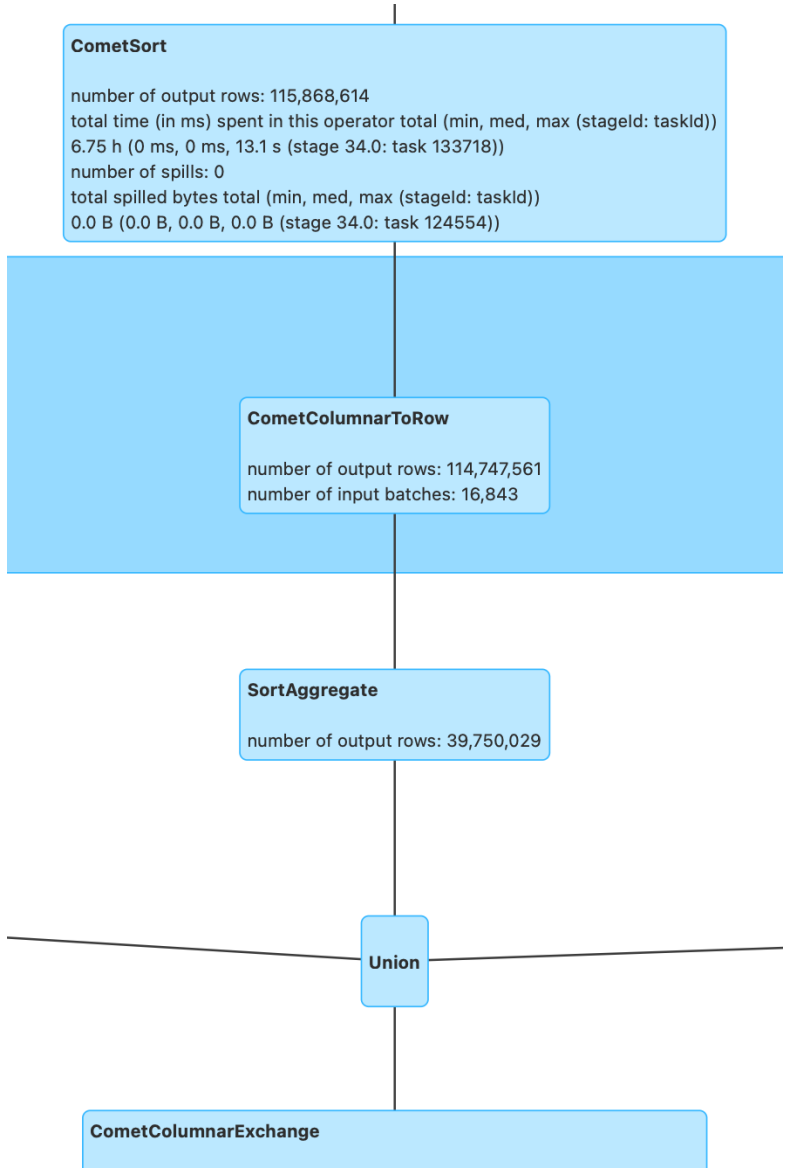
Вывод: Apache DataFusion Comet версии 0.9 поддерживает **128 разных выражений и функций**, распределённых по 15 категориям.

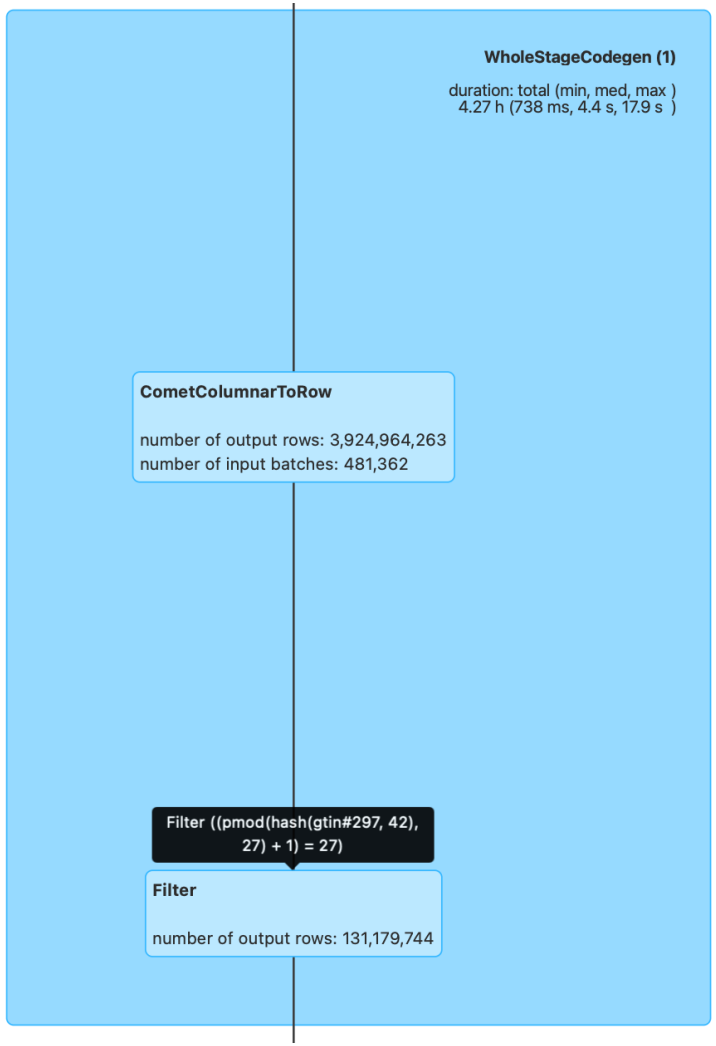
Если нужно будет покатегорийный список или подробнее про какую-то функцию — дайте знать!

Источники

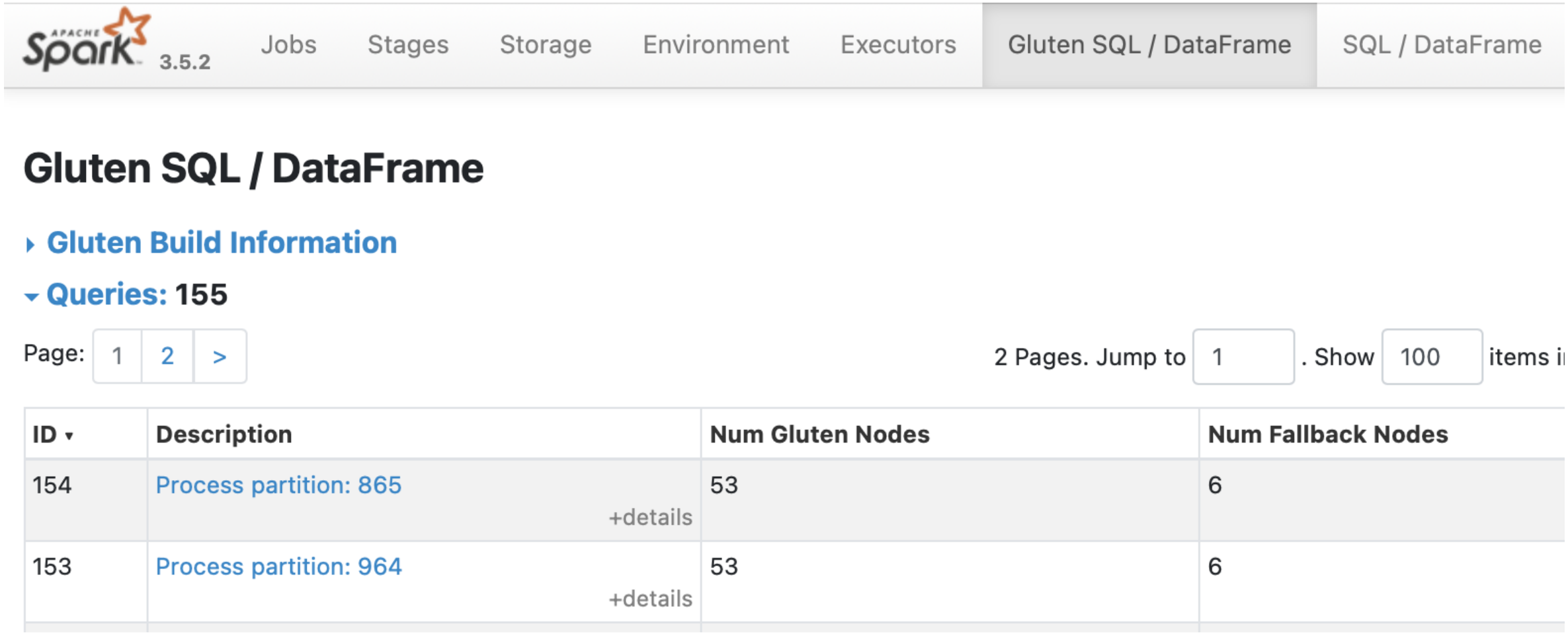
Fallback


VeloxColumnarToRow
number of output rows: 12,437,993,821
number of input batches: 4,088,191
time to convert total (min, med, max)
8.29 h (423 ms, 920 ms, 3.9 s)
WholeStageCodegen (1)
duration: total (min, med, max)
231.06 h (9.4 s, 23.1 s, 1.5 m)
Project
RowToVeloxColumnar
number of input rows: 12,437,993,821
number of output batches: 3,051,021
time to convert total (min, med, max)
16.60 h (928 ms, 1.7 s, 10.4 s)


CometSort
number of output rows: 115,868,614
total time (in ms) spent in this operator total (min, med, max (stageId: taskId))
6.75 h (0 ms, 0 ms, 13.1 s (stage 34.0: task 133718))
number of spills: 0
total spilled bytes total (min, med, max (stageId: taskId))
0.0 B (0.0 B, 0.0 B, 0.0 B (stage 34.0: task 124554))
CometColumnarToRow
number of output rows: 114,747,561
number of input batches: 16,843
SortAggregate
number of output rows: 39,750,029
Union
CometColumnarExchange


WholeStageCodegen (1)
duration: total (min, med, max)
4.27 h (738 ms, 4.4 s, 17.9 s)
CometColumnarToRow
number of output rows: 3,924,964,263
number of input batches: 481,362
Filter ((pmod(hash(gtin#297, 42), 27) + 1) = 27)
Filter
number of output rows: 131,179,744

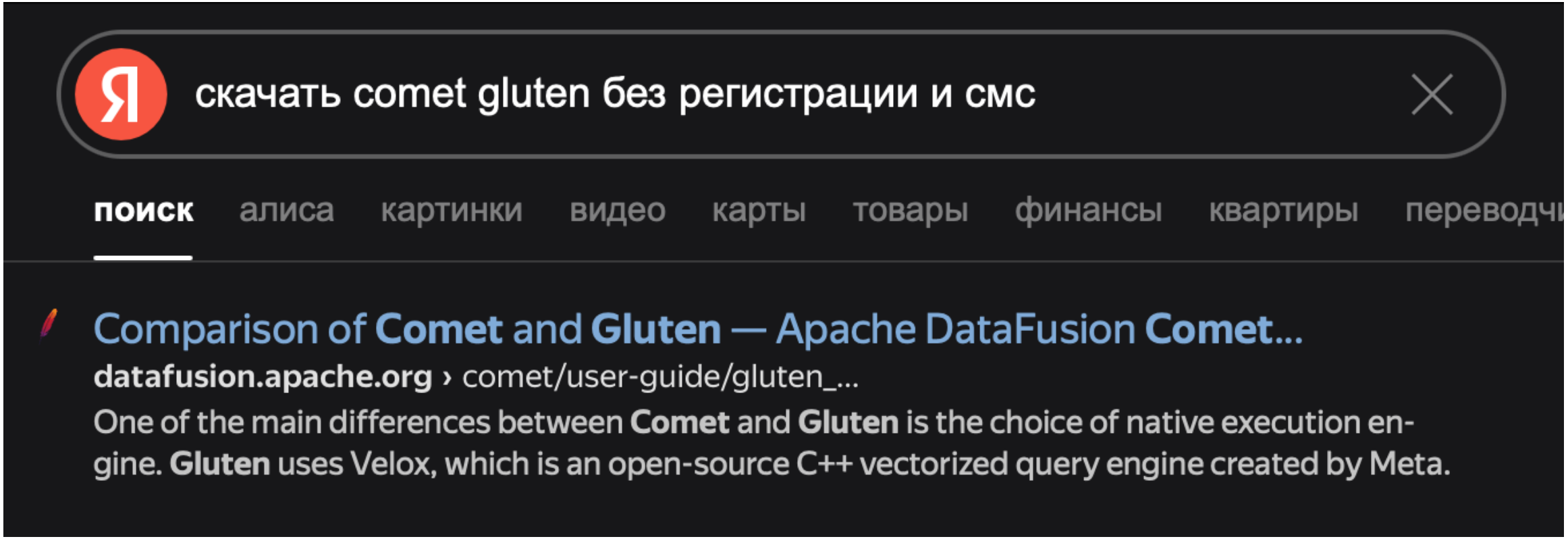
Fallback


Spark 3.5.2
Jobs
Stages
Storage
Environment
Executors
Gluten SQL / DataFrame
SQL / DataFrame
Gluten SQL / DataFrame
Gluten Build Information
Queries: 155
Page: 1 2 >
2 Pages. Jump to 1 . Show 100 items i
ID
Description
Num Gluten Nodes
Num Fallback Nodes
154
Process partition: 865
+details
53
6
153
Process partition: 964
+details
53
6

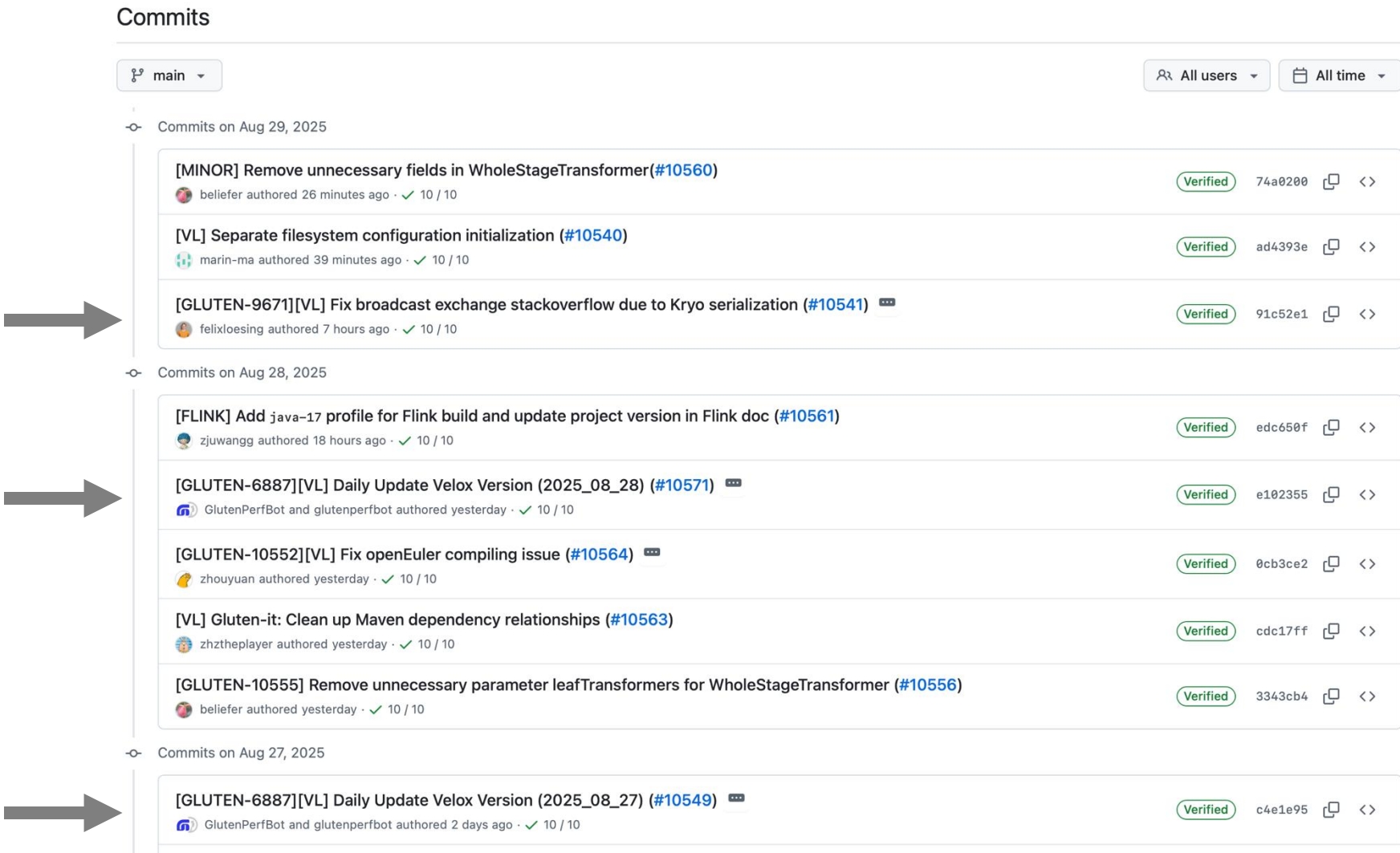
Как это всё заставить работать?

Сборка

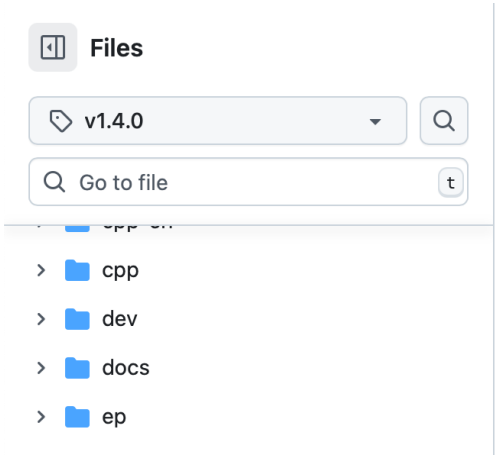
Сборка – версии Spark



Сборка – daily updates



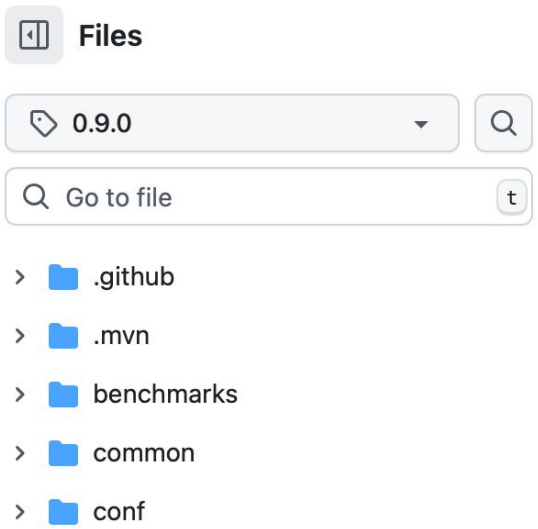
Сборка – версии Spark



incubator-gluten / pom.xml

Code | Blame | 1670 lines (1651 loc) · 59.7 KB

```
347         <id>spark-3.5</id>
348         <properties>
349           <sparkbundle.version>3.5</sparkbundle.version>
350           <sparkshim.artifactId>spark-sql-columnar-shims-spark35</sparkshim.artifactId>
351           <spark.version>3.5.2</spark.version>
352           <iceberg.version>1.5.0</iceberg.version>
353           <delta.package.name>delta-spark</delta.package.name>
354           <delta.version>3.2.0</delta.version>
355           <delta.binary.version>32</delta.binary.version>
356           <hudi.version>0.15.0</hudi.version>
```



datafusion-comet / pom.xml

Code | Blame | 1146 lines (1093 loc) · 40.8 KB

```
600         <profile>
601           <id>spark-3.5</id>
602           <properties>
603             <scala.version>2.12.18</scala.version>
604             <spark.version>3.5.6</spark.version>
605             <spark.version.short>3.5</spark.version.short>
606             <parquet.version>1.13.1</parquet.version>
607             <slf4j.version>2.0.7</slf4j.version>
608             <shims.minorVerSrc>spark-3.5</shims.minorVerSrc>
609           </properties>
610         </profile>
```

Сборка – QAT + IAA

Intel® QuickAssist Technology (QAT) support

Gluten supports using Intel® QuickAssist Technology (QAT) for data compression during Spark Shuffle. It benefits from QAT Hardware-based acceleration on compression/decompression, and uses Gzip as compression format for higher compression ratio to reduce the pressure on disks and network transmission.

This feature is based on QAT driver library and QATzip library. Please manually download QAT driver for your system, and follow its README to build and install on all Driver and Worker node: Intel® QuickAssist Technology Driver for Linux* – HW Version 2.0.

Intel® In-memory Analytics Accelerator (IAA/IAX) support

Similar to Intel® QAT, Gluten supports using Intel® In-memory Analytics Accelerator (IAA, also called IAX) for data compression during Spark Shuffle. It benefits from IAA Hardware-based acceleration on compression/decompression, and uses Gzip as compression format for higher compression ratio to reduce the pressure on disks and network transmission.

This feature is based on Intel® QPL.

Сборка – Comet


```
25/07/21 15:30:09 ERROR Executor: Exception in task 2.1 in stage 7.0 (TID 19)
org.apache.comet.CometNativeException: General execution error with reason: Generic HadoopFileSystem error: Hdfs support is not enabled in this build.
   at org.apache.comet.Native.executePlan(Native Method)
   at org.apache.comet.CometExecIterator.$anonfun$getNextBatch$2(CometExecIterator.scala:155)
   at org.apache.comet.CometExecIterator.$anonfun$getNextBatch$2$adapted(CometExecIterator.scala:154)
   at org.apache.comet.vector.NativeUtil.getNextBatch(NativeUtil.scala:157)
   at org.apache.comet.CometExecIterator.$anonfun$getNextBatch$1(CometExecIterator.scala:154)
   at org.apache.comet.Tracing$.withTrace(Tracing.scala:31)
   at org.apache.comet.CometExecIterator.getNextBatch(CometExecIterator.scala:152)
   at org.apache.comet.CometExecIterator.hasNext(CometExecIterator.scala:203)
   at org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIteratorForCodegenStage4.cometcolumnartorow_nextBatch_0$(Unknown Source)
   at org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIteratorForCodegenStage4.processNext(Unknown Source)
   at org.apache.spark.sql.execution.BufferedRowIterator.hasNext(BufferedRowIterator.java:43)
   at org.apache.spark.sql.execution.WholeStageCodegenEvaluatorFactory$WholeStageCodegenPartitionEvaluator$$anon$1.hasNext(WholeStageCodegenEvaluatorFa
   at org.apache.spark.sql.execution.SparkPlan.$anonfun$getByteArrayRdd$1(SparkPlan.scala:388)
   at org.apache.spark.rdd.RDD.$anonfun$mapPartitionsInternal$2(RDD.scala:893)
```

Сборка - версии Spark

Сборка – версии Spark

Сборка – версии Spark

APACHE DATAFUSION COMET™

```
./mvnw clean package -Prelease -Pjdk17 -DskipTests \
-Dskip.surefire.tests=true -Dscala.version=2.12.15 -Dspark.version=3.5.1 \
-Dspark.version.short=3.5 -Dparquet.version=1.13.1 -Dslf4j.version=2.0.7 \
-Dshims.minorVerSrc=spark-3.5 -Dmaven.gitcommitid.skip=true;
```

```
[INFO]
[INFO] --- scala:4.8.0:compile (scala-compile-first) @ comet-common-spark3.5_2.12 ---
[INFO] Compiler bridge file: /root/.sbt/1.0/zinc/org.scala-sbt/org.scala-sbt-compiler-bridge_2.12-1.8.0-bin_2.12.15__61.0-1.8.0_20221110T195421.jar
[INFO] compiling 19 Scala sources and 44 Java sources to /opt/anst/external/comet/common/target/classes ...
[ERROR] [Error] /opt/anst/external/comet/common/src/main/spark-3.5/org/apache/spark/sql/comet/shims/ShimTaskMetrics.scala:30: value withExternalAccums is not a member of org.apache.spark.executor.TaskMetrics
[ERROR] one error found
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for Comet Project Parent POM 0.9.0:
[INFO]
[INFO] Comet Project Parent POM ........................... SUCCESS [  1.482 s]
[INFO] comet-common ....................................... FAILURE [ 10.113 s]
[INFO] comet-spark ........................................ SKIPPED
[INFO] comet-spark-integration ............................ SKIPPED
[INFO] comet-fuzz ......................................... SKIPPED
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  11.741 s
[INFO] Finished at: 2025-08-27T21:03:26+03:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal net.alchim31.maven:scala-maven-plugin:4.8.0:compile (scala-compile-first) on project comet-common-spark3.5_2.12: Execution scala-compile-first of goal net.alchim31.maven:scala-ma
ocess exited with an error: 255 (Exit value: 255) -> [Help 1]
[ERROR]
```

Сборка – версии Spark


```
mvn clean compile -Pbackends-velox -Pjava-17 -Pspark-3.5 \
-Phadoop-3.3 -Dspark.version=3.5.5 -DskipTests
```

```
[INFO] --- scala-maven-plugin:4.8.0:compile (scala-compile-first) @ spark-sql-columnar-shims-spark35 ---
[WARNING]  Expected all dependencies to require Scala version: 2.12.15
[WARNING]  org.apache.spark:spark-core_2.12:3.5.5 requires scala version: 2.12.18
[WARNING] Multiple versions of scala libraries detected!
[INFO] /opt/anst/external/gluten/shims/spark35/src/main/java:-1: info: compiling
[INFO] /opt/anst/external/gluten/shims/spark35/src/main/scala:-1: info: compiling
[INFO] Compiling 24 source files to /opt/anst/external/gluten/shims/spark35/target/classes at 1756318647486
[INFO] compiler plugin: BasicArtifact(org.wartremover,wartremover_2.12,3.1.6,null)
[ERROR] /opt/anst/external/gluten/shims/spark35/src/main/scala/org/apache/gluten/sql/shims/spark35/Spark35Shims.scala:152: error: not enough arguments for method getPartitionedFile: (file: or
.Path, partitionValues: org.apache.spark.sql.catalyst.InternalRow)org.apache.spark.sql.execution.datasources.PartitionedFile.
[ERROR] Unspecified value parameter partitionValues.
[ERROR]       .flatMap(p => p.files.map(f => PartitionedFileUtil.getPartitionedFile(f, p.values)))
[ERROR]                                                          ^
[ERROR] /opt/anst/external/gluten/shims/spark35/src/main/scala/org/apache/gluten/sql/shims/spark35/Spark35Shims.scala:418: error: not enough arguments for method splitFiles: (sparkSession: or
hMetadata, filePath: org.apache.hadoop.fs.Path, isSplitable: Boolean, maxSplitBytes: Long, partitionValues: org.apache.spark.sql.catalyst.InternalRow)Seq[org.apache.spark.sql.execution.datasc
[ERROR] Unspecified value parameter partitionValues.
[ERROR]     PartitionedFileUtil.splitFiles(
[ERROR]                         ^
[ERROR] /opt/anst/external/gluten/shims/spark35/src/main/scala/org/apache/spark/sql/execution/AbstractFileSourceScanExec.scala:184: error: not enough arguments for method getPartitionedFile:
hadoop.fs.Path, partitionValues: org.apache.spark.sql.catalyst.InternalRow)org.apache.spark.sql.execution.datasources.PartitionedFile.
[ERROR] Unspecified value parameter partitionValues.
[ERROR]         .flatMap(p => p.files.map(f => PartitionedFileUtil.getPartitionedFile(f, p.values)))
[ERROR]                                                            ^
[ERROR] /opt/anst/external/gluten/shims/spark35/src/main/scala/org/apache/spark/sql/execution/AbstractFileSourceScanExec.scala:272: error: not enough arguments for method splitFiles: (sparkSe
StatusWithMetadata, filePath: org.apache.hadoop.fs.Path, isSplitable: Boolean, maxSplitBytes: Long, partitionValues: org.apache.spark.sql.catalyst.InternalRow)Seq[org.apache.spark.sql.executi
[ERROR] Unspecified value parameter filePath.
[ERROR]                 PartitionedFileUtil.splitFiles(
[ERROR]                                     ^
[ERROR] four errors found
[ERROR] exception compilation error occurred!!!
org.apache.commons.exec.ExecuteException: Process exited with an error: 1 (Exit value: 1)
    at org.apache.commons.exec.DefaultExecutor.executeInternal (DefaultExecutor.java:404)
```

Сборка – версии Spark


```
Caused by: java.lang.NoSuchMethodError: 'java.util.Map
org.apache.spark.shuffle.IndexShuffleBlockResolver$.$lessinit$greater$default$3()'
at org.apache.spark.shuffle.sort.ColumnarShuffleManager.<init>
(ColumnarShuffleManager.scala:38)
```

Рабочий сетап

Comet - 0.9.0 (self built) + Spark 3.5.5 Java 17

Gluten – 1.4.0 (с сайта) + Spark 3.5.2 Java 17

Часть на rust/c++

Сборка


Java Setup - Progress
Java
ORACLE
Status: Registering Java Runtime Environment
3 Billion Devices Run Java
Computers, Printers, Routers, Cell Phones, BlackBerry,
Kindle, Parking Meters, Public Transportation Passes, ATMs,
Credit Cards, Home Security Systems, Cable Boxes, TVs...
ORACLE

Сборка – rust

```
#
# A fatal error has been detected by the Java Runtime Environment:
#
#  SIGILL (0x4) at pc=0x00007f7177d003c0, pid=383814, tid=384933
#
# JRE version: Java(TM) SE Runtime Environment (17.0.4.1+1) (build 17.0.4.1+1-LTS-2)
# Java VM: Java HotSpot(TM) 64-Bit Server VM (17.0.4.1+1-LTS-2, mixed mode, sharing, tiered, compressed oops, compressed class ptrs, g1 gc, linux-amd64)
# Problematic frame:
# C  [libcomet-9515330560790993069.so+0x32113c0]  arrow_buffer::util::bit_chunk_iterator::UnalignedBitChunk::count_ones::h3c1a9aad0bd49324+0x50
#
# Core dump will be written. Default location: Core dumps may be processed with "/usr/share/apport/apport -p%p -s%s -c%c -d%d -P%P -u%u -g%g -- %E" (or dur
#
# An error report file with more information is saved as:
```

Сборка – rust

datafusion-comet / **Makefile**

Code | Blame | 109 lines (99 loc) · 5.39 KB

```
100     release:
101         cd native && RUSTFLAGS="$(RUSTFLAGS) -Ctarget-cpu=native" cargo build --release $(FEATURES_ARG)
102         ./mvnw install -Prelease -DskipTests $(PROFILES)
103     release-nogit:
104         cd native && RUSTFLAGS="-Ctarget-cpu=native" cargo build --release
105         ./mvnw install -Prelease -DskipTests $(PROFILES) -Dmaven.gitcommitid.skip=true
106     benchmark-%: release
107         cd spark && COMET_CONF_DIR=$(shell pwd)/conf MAVEN_OPTS='-Xmx20g ${call spark_jvm_17_extra_args}' .
```

Сборка – rust

Executors

Show All entries

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks
48	h0[redacted]41357	Dead	0	228.5 KiB / 114.9 GiB	0.0 B	40	0	40	8
64	h9[redacted]44125	Dead	0	228.5 KiB / 114.9 GiB	0.0 B	40	0	40	5
73	h9[redacted]34929	Dead	0	228.5 KiB / 114.9 GiB	0.0 B	40	0	40	4
101	h9[redacted]45487	Dead	0	84.5 KiB / 114.9 GiB	0.0 B	40	0	40	17
driver	h11[redacted]:33965	Active	0	7.5 MiB / 28.5 GiB	0.0 B	0	0	0	0
1	h5[redacted]:35275	Active	0	5.2 MiB / 114.9 GiB	0.0 B	40	40	0	294

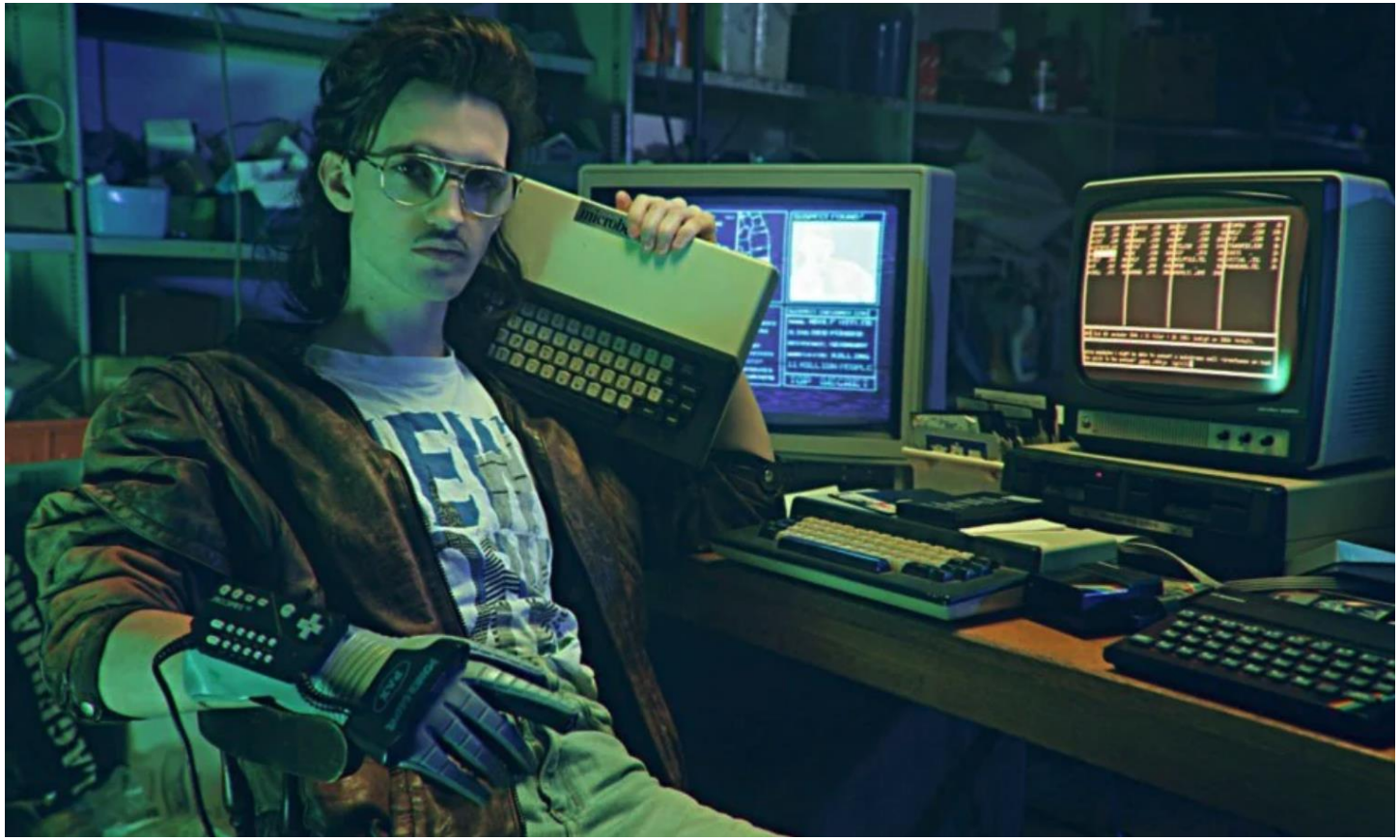
Сборка – rust

```
root[redacted]/n.blagodarny# rustc --print target-cpus
Available CPUs for this target:
    native                  - Select the CPU of the current host (currently icelake-server).
    alderlake
    amdfam10
    arrowlake
    arrowlake-s
    arrowlake_s
    athlon
    athlon-4
    athlon-fx
    athlon-mp
    athlon-tbird
    athlon-xp
    athlon64
    athlon64-sse3
    atom
    atom_sse4_2
    atom_sse4_2_movbe
    barcelona
    bdver1
    bdver2
    bdver3
    bdver4
    bonnell
    broadwell
    btver1
    btver2
    c3
    c3-2
    cannonlake
    cascadelake
    clearwaterforest
    cooperlake
    core-avx-i
    core-avx2
    core2
    core_2_duo_sse4_1
    core_2_duo_ssse3
    core_2nd_gen_avx
    core_3rd_gen_avx
    core_4th_gen_avx
    core_4th_gen_avx_tsx
    core_5th_gen_avx
    core_5th_gen_avx_tsx
    core_aes_pclmulqdq
    core_i7_sse4_2
    corei7
    corei7-avx
```

```
    pentium
    pentium-m
    pentium-mmx
    pentium2
    pentium3
    pentium3m
    pentium4
    pentium4m
    pentium_4
    pentium_4_sse3
    pentium_ii
    pentium_iii
    pentium_iii_no_xmm_regs
    pentium_m
    pentium_mmx
    pentium_pro
    pentiumpro
    prescott
    raptorlake
    rocketlake
    sandybridge
    sapphirerapids
    sierraforest
    silvermont
    skx
    skylake
    skylake-avx512
    skylake_avx512
    slm
    tigerlake
    tremont
    westmere
    winchip-c6
    winchip2
    x86-64                  - This is the default target CPU for the current build target
    x86-64-v2
    x86-64-v3
    x86-64-v4
    yonah
    znver1
    znver2
    znver3
    znver4
    znver5
```

Сборка – rust

```
/home/n.blagodarny# rustc -C target-feature=+avx2,+fma
```



C++ Gluten

`spark.yarn.exclude.nodes`	(none)	Comma-separated list of YARN node names which are excluded from resource allocation.

Classpath

java.lang.ClassNotFoundException: org.apache.spark.sql.comet.execution.shuffle.CometShuffleManager #864

Open

radhikabajaj123 opened on Aug 22, 2024 · edited by radhikabajaj123 — Edits

Hello,

I am getting the following exception when running spark-submit:

```
Exception in thread "main" java.lang.reflect.UndeclaredThrowableException
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1780)
at org.apache.spark.deploy.SparkHadoopUtil.runAsSparkUser(SparkHadoopUtil.scala:67)
at org.apache.spark.executor.CoarseGrainedExecutorBackend$.run(CoarseGrainedExecutorBackend.scala:429)
at org.apache.spark.executor.YarnCoarseGrainedExecutorBackend$.main(YarnCoarseGrainedExecutorBackend.scala:83)
at org.apache.spark.executor.YarnCoarseGrainedExecutorBackend.main(YarnCoarseGrainedExecutorBackend.scala)
Caused by: java.lang.ClassNotFoundException: org.apache.spark.sql.comet.execution.shuffle.CometShuffleManager
```

Assignees: No one assigned

Labels: No labels

Type: No type

Projects: No projects

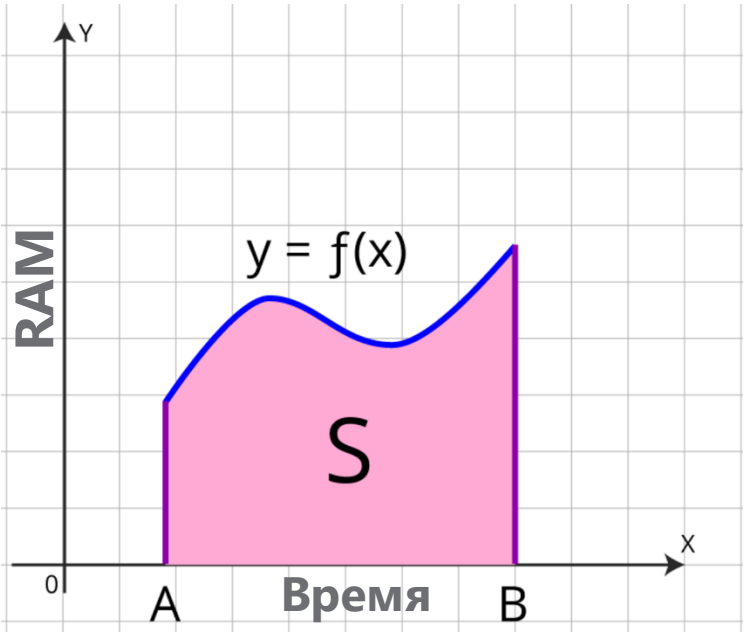
Run Spark Shell with Comet enabled

Make sure `SPARK_HOME` points to the same Spark version as Comet was built for.

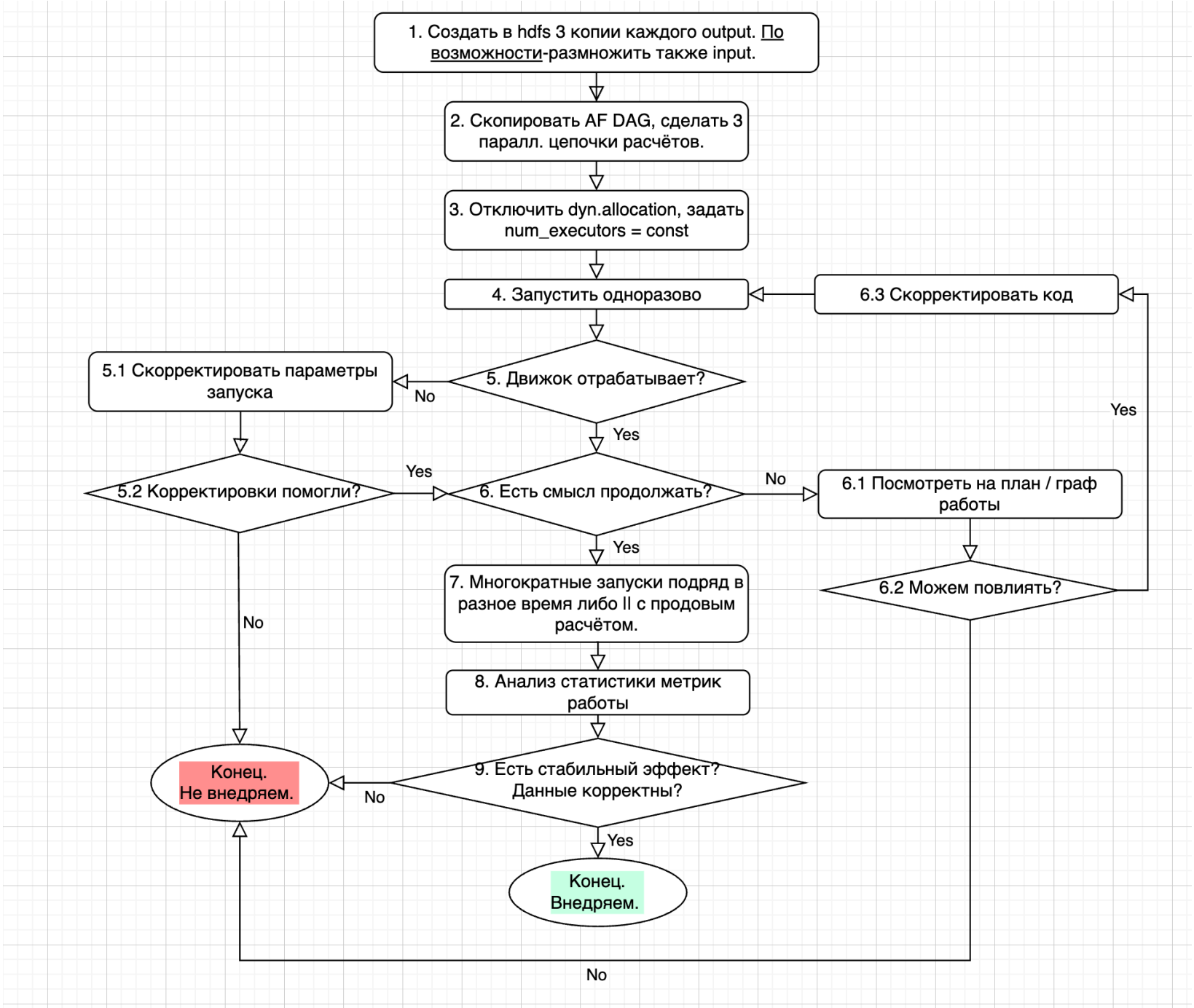
```
export COMET_JAR=spark/target/comet-spark-spark3.5_2.12-0.10.0-SNAPSHOT.jar

$SPARK_HOME/bin/spark-shell \
    --jars $COMET_JAR \
    --conf spark.driver.extraClassPath=$COMET_JAR \
    --conf spark.executor.extraClassPath=$COMET_JAR \
    --conf spark.plugins=org.apache.spark.CometPlugin \
```

Цели и подходы к тестированию

Зачем?

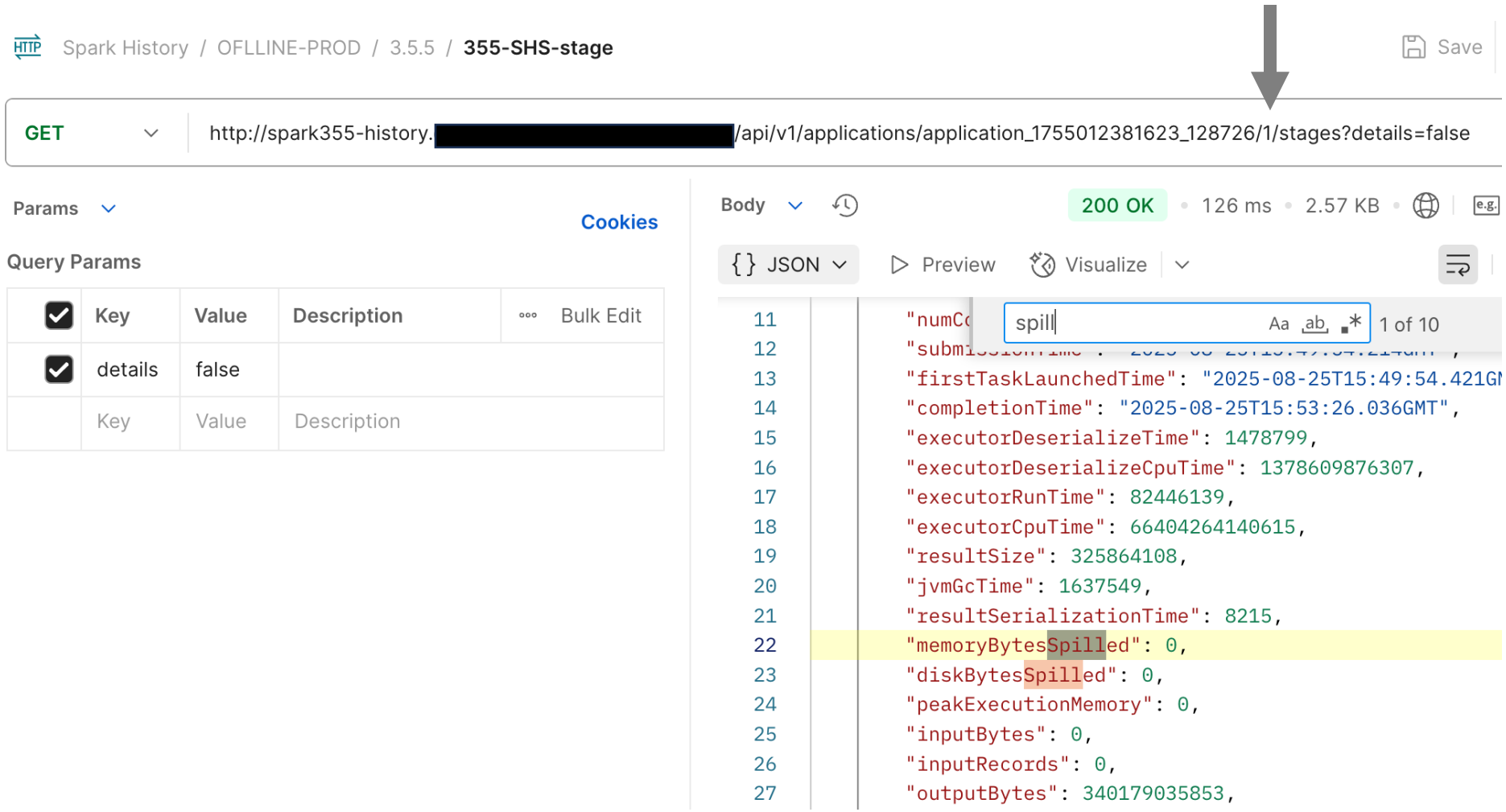


Как?

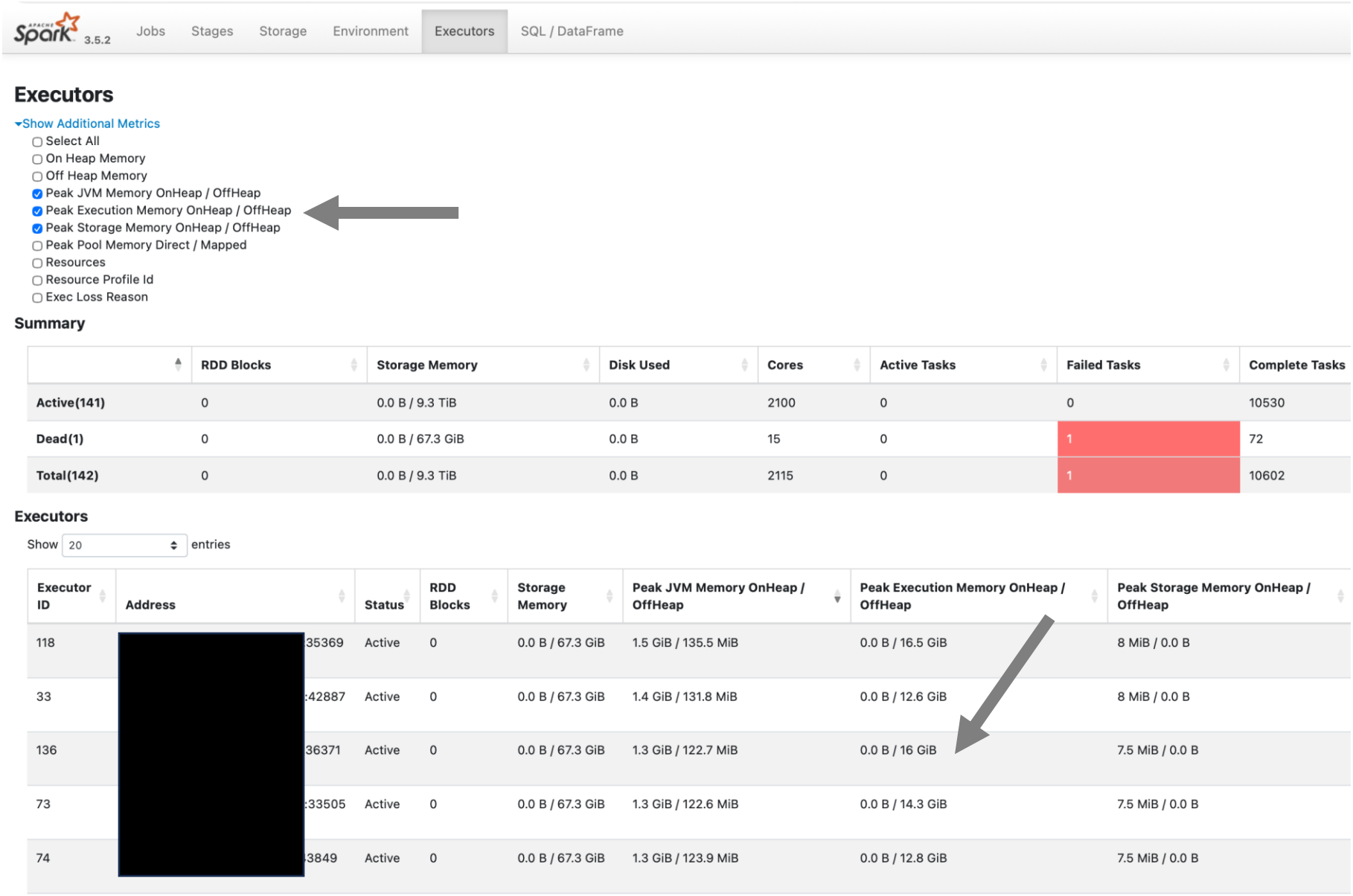

1. Создать в hdfs 3 копии каждого output. По возможности-размножить также input.
2. Скопировать AF DAG, сделать 3 паралл. цепочки расчётов.
3. Отключить dyn.allocation, задать num_executors = const
4. Запустить одноразово
6.3 Скорректировать код
5. Движок отрабатывает?
No
5.1 Скорректировать параметры запуска
Yes
Yes
5.2 Корректировки помогли?
Yes
6. Есть смысл продолжать?
No
6.1 Посмотреть на план / граф работы
Yes
6.2 Можем повлиять?
Yes
No
7. Многократные запуски подряд в разное время либо II с продовым расчётом.
8. Анализ статистики метрик работы
Конец.
Не внедряем.
No
9. Есть стабильный эффект? Данные корректны?
Yes
Конец.
Внедряем.
No

Лайфхаки

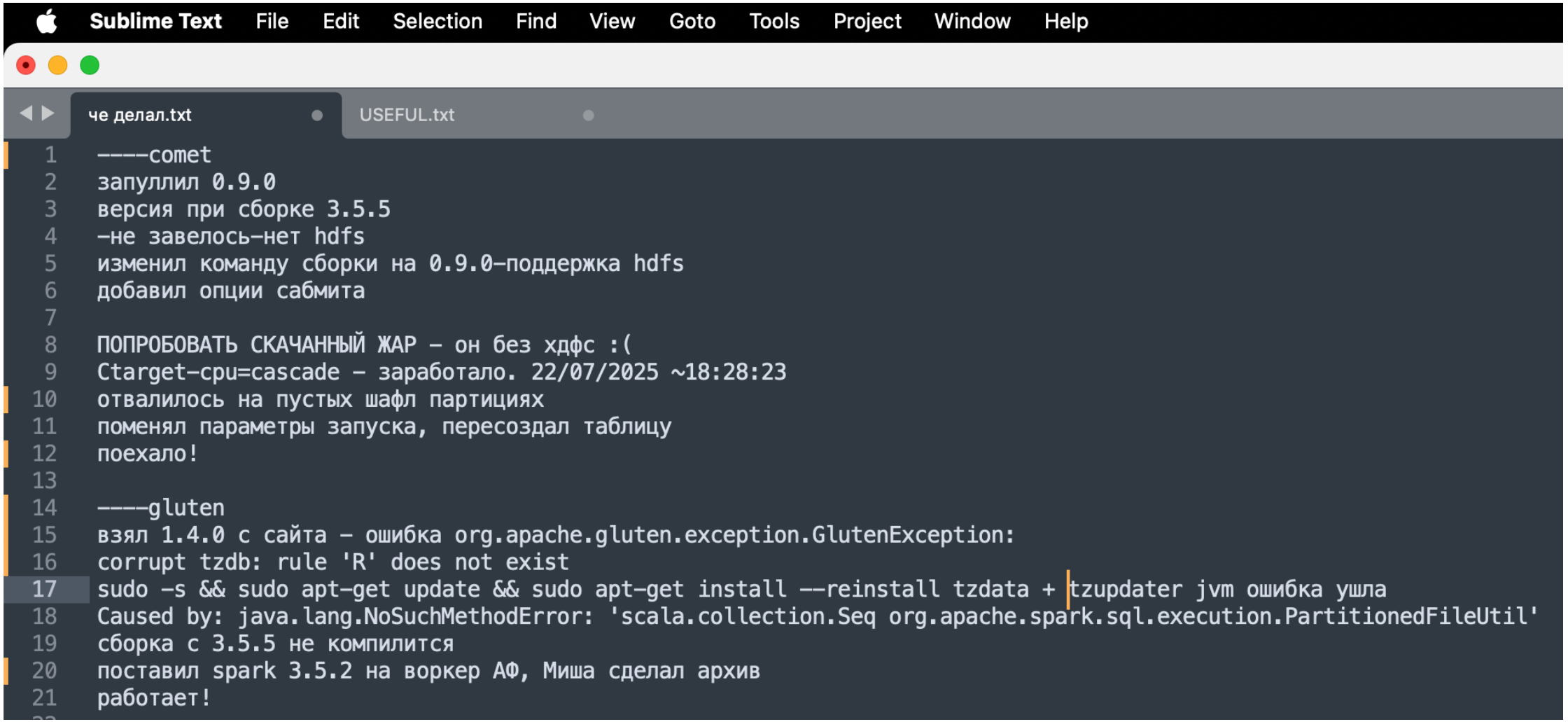
Мониторинг spill



Мониторинг нативной RAM



Логирование действий-инфра



Логирование действий-параметры

			Gb			
			jvm + offheap	non jvm	spill	время мин
	1	orig	58+8		23,3	38
		comet	4	30	199,9	55
		gluten	8	30	5 826,6	103
	2	orig	58+8		80,4	35
		comet	8	30	129,0	52
		gluten	10	35	5 652,5	100
	3	orig	58+8		0,0	39
		comet	10	30	117,0	55
		gluten	20	30	5 867,5	118
	4	orig	58+8		0,0	33
		comet	20	20	46,5	67
		gluten	25	20	6 789,1	167
	5	orig	58+8		0,0	30
		comet	20	20	43,0	49
		gluten	25	20	6 195,2	107
	6	orig	58+8		0,0	38
		comet	20	20	35,0	45
убрали SHJ		gluten	25	20	870,0	122

Результаты тестов

Job 1 – Silver

25 партиций
По 5 parallel job

```scala
val df1 = someHdfsInput1 // 31 999 files , 27 535 079 320 rows, 3.7 TiB
  .filter()    // x3 - take 1 part/1 subpart + field isIn + or + case
  .transform() // case when when when else
  .select()    // 31 fields + 3 case when

val df2 = someHdfsInput2 // 31 999 files , 6 965 333 811 rows, 1.98 TiB
  .filter()     // x1 - take 1 part/1 subpart
  .transform()  //case when
  .select()     // 38 columns + 2 case when

val df3 = someHdfsInput3 // 31 999 files , 946 949 479 rows, 136 GiB
  .filter()     //x1 - take 1 part/1 subpart + 1 col === const
  .select()     // 3 columns

val df4 = df1.join(df2, Seq("key_col")).join(df3, Seq("key_col"), "left")

val windowDesc                = Window.partitionBy("key_col").orderBy($"dttm".desc)
val windowAsc                 = Window.partitionBy("key_col").orderBy($"dttm")
val windowAsc1                = Window.partitionBy("key_col").orderBy($"bool_col", $"dttm")
val windowAsc2                = Window.partitionBy("key_col").orderBy($"dttm")
val windowAscUnbounded        = windowAsc.rowsBetween(Window.unboundedPreceding, Window.unboundedFollowing)
val windowAscUnboundedMinusOne = windowAsc.rowsBetween(Window.unboundedPreceding, -1)

val collect_list_func         = collect_list(when($"col_1" === 69, $"col2")).over(windowAscUnboundedMinusOne)
val map_from_arr_func         = map_from_arrays(
  collect_list(when(someCond1, $"col_2")).over(windowDesc),
  collect_list(when(someCond1, struct(someColSeq.map(col): _*))).over(windowDesc)
)

df4.withColumn()  // x7  case when and/or coalesce
    .withColumn()  // x1  cast
    .withColumn()  // x7  with array functions
    .filter()      // x1  and/or/not < > !=
    .withColumn()  // x10 case when and/or coalesce
    .withColumn()  // x5  over window
    .filter()      // x1  and/or/not < > != using window results
    .withColumn()  // x13 case when and/or coalesce
    .withColumn()  // x3 calc over window
    .select($"col_1", $"col_2")   // 85 columns
    .write.parquet("hdfs://")
```

Job 1 – Silver

	Exec.mem (JVM + Native)
No engine *3,3 часа*	**128 Gb**
Comet *3,6 часа*	**128 +128 Gb**
Velox *11 часов*	**128 +128 Gb**

Job Id ▲	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	Process partition: 7 Process partition: 7	2025/09/02 15:19:43	12 min	1/1	31999/31999

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (58)
+- WriteFiles (57)
   +- * Sort (56)
      +- * Project (55)
         +- * Project (54)
            +- * Project (53)
```

2	Process partition: 7 Process partition: 7	2025/09/02 15:19:42	22 min	1/1	31999/31999

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (58)
+- WriteFiles (57)
   +- * Sort (56)
      +- * Project (55)
         +- * Project (54)
            +- * Project (53)
```

Job Id ▲	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	Process partition: 7 Process partition: 7	2025/09/02 15:19:43	21 min	1/1	31999/31999 (40 failed)

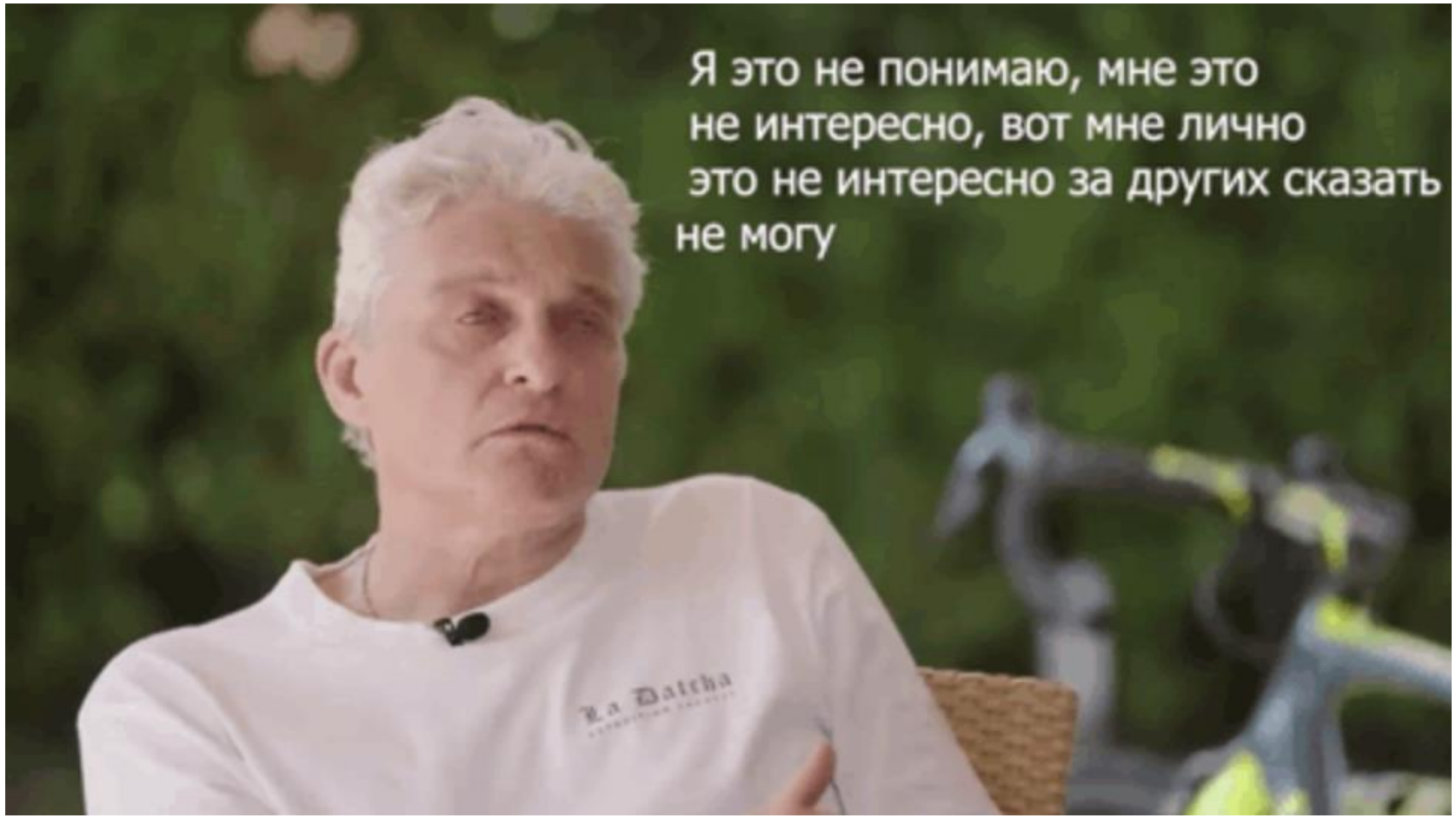
▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (83)
+- WriteFiles (82)
   +- VeloxColumnarToRow (81)
      +- ^ ProjectExecTransformer (79)
         +- ^ SortExecTransformer (78)
            +- ^ ProjectExecTransformer (77)
```

```
map_from_arrays(_we0#6889,
```

SmartData 2025

Job 1 – Silver



«ДАННОЕ ЛИЦО ВКЛЮЧЕНО В РЕЕСТР ИНОСТРАННЫХ АГЕНТОВ МИНИСТЕРСТВА ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Job 2 – Gold

```scala
// files: 19 374 size: 889.5 GiB rows: 11 954 241 420 partitions: 54
val df1 = spark.read.parquet("/foo/bar/silver")

val skipGroupColumns = Array("col_1","col_2", ...) //  14 columns
val cols = df1.columns.filterNot(skipGroupColumns.contains(_)).map(col)

val aggDf = df1.groupBy(cols: _*)
  .agg(
    sum("col_1").as("col_1")
    ,sum("col_2").as("col_2")
    ,sum("col_3").as("col_3")
    ,sum("col_4").as("col_4")
    ,max("key_example").as("key_example")
    ,max("ts").as("ts")
  )
  .withColumn("ts_insert", current_timestamp())
  .repartition($"shard_num", monotonically_increasing_id() % 600)
  .sortWithinPartitions() // 6 columns
  .write.mode("overwrite")
  .partitionBy("shard_num")
  .parquet("/foo/bar/gold")

```

Job 2 – Gold

	Exec.mem (JVM + Native)
No engine *15 мин*	**82 Gb**
Comet *56 мин*	**80+40 Gb**
Velox *1,4 часа*	**80+80 Gb**

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	save [redacted] save	2025/09/08 17:24:08	15 min	3/3	33875/33875
0	load [redacted] load	2025/09/08 17:24:03	2 s	1/1	1/1

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (12)
+- WriteFiles (11)
   +- * Sort (10)
      +- * Project (9)
         +- Exchange (8)
            +- SortAggregate (7)
```

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	save [redacted] save	2025/09/08 17:24:13	56 min	3/3	33875/33875 (13 failed)
0	load [redacted] load	2025/09/08 17:24:07	2 s	1/1	1/1

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (15)
+- WriteFiles (14)
   +- * CometColumnarToRow (13)
      +- CometSort (12)
         +- CometProject (11)
```

SortAggregate

number of output rows: 9,816,517,450

▾ Completed Jobs (1)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	load [redacted] load	2025/09/08 17:24:10	2 s	1/1	1/1

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

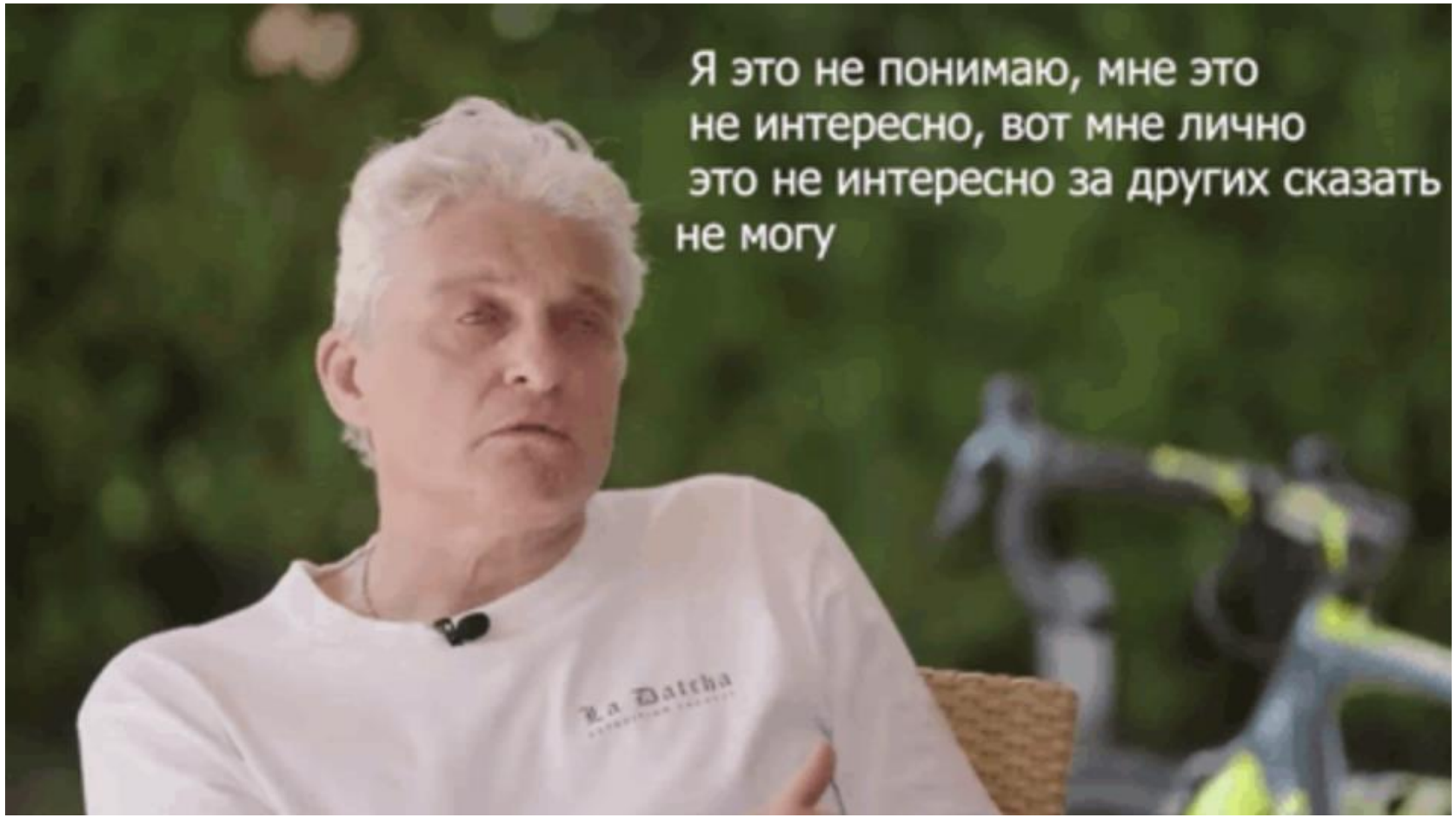
▾ Failed Jobs (1)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	save [redacted] save	2025/09/08 17:24:16	1.4 h	2/3 (1 failed)	64319/73873 (3637 failed)

```
Caused by: org.apache.gluten.exception.GlutenException:
Error Source: RUNTIME
Error Code: INVALID_STATE
Reason: An unsupported nested encoding was found.
```

Job 1 – Silver



«ДАННОЕ ЛИЦО ВКЛЮЧЕНО В РЕЕСТР ИНОСТРАННЫХ АГЕНТОВ МИНИСТЕРСТВА ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Job 3 – Gold

28 партиций
По 5 parallel job

```
82  //files: 500 rows: 635 832 198 cols: 93
83  val df = spark.read.parquet("/foo/bar/gold").where("size(col_1) = 0")
84
85  val res = df
86    .filter($"shard_num" === 1)
87    .withColumn("")      // x2 simple case when
88    .where("c_t != -1")
89    .groupBy()           // 3x case when + 23 columns
90    .agg()               // 2x sum + 2 max()
91    .withColumn("ts_ins", current_timestamp())
92    .drop("colname")
93    .repartition($"col", 600)
94    .write.mode("overwrite")
95    .parquet("/foo/bar/gold/agg")
96
```

Job 3 – Gold

No engine

20 мин

24	Process partition: 24 Process partition: 24	2025/09/12 11:14:50	3.4 min	3/3	5600/5600

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (18)
+- WriteFiles (17)
   +- * Project (16)
      +- * Sort (15)
         +- * Project (14)
            +- Exchange (13)
```

Comet

19 мин

24	Process partition: 24 Process partition: 24	2025/09/12 11:13:46	3.2 min	3/3	3700/3700

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (18)
+- WriteFiles (17)
   +- * CometColumnarToRow (16)
      +- CometColumnarExchange (15)
         +- SortAggregate (14)
            +- * CometColumnarToRow (13)
```

Velox

20.5 мин

24	Process partition: 24 Process partition: 24	2025/09/12 11:15:44	2.7 min	3/3	3700/3700

▾ Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (31)
+- VeloxColumnarWriteFiles (30)
   :- ^ WriteFilesExecTransformer (25)
   :  +- ^ InputIteratorTransformer (24)
   :     +- ColumnarExchange (22)
   :        +- VeloxResizeBatches (21)
```

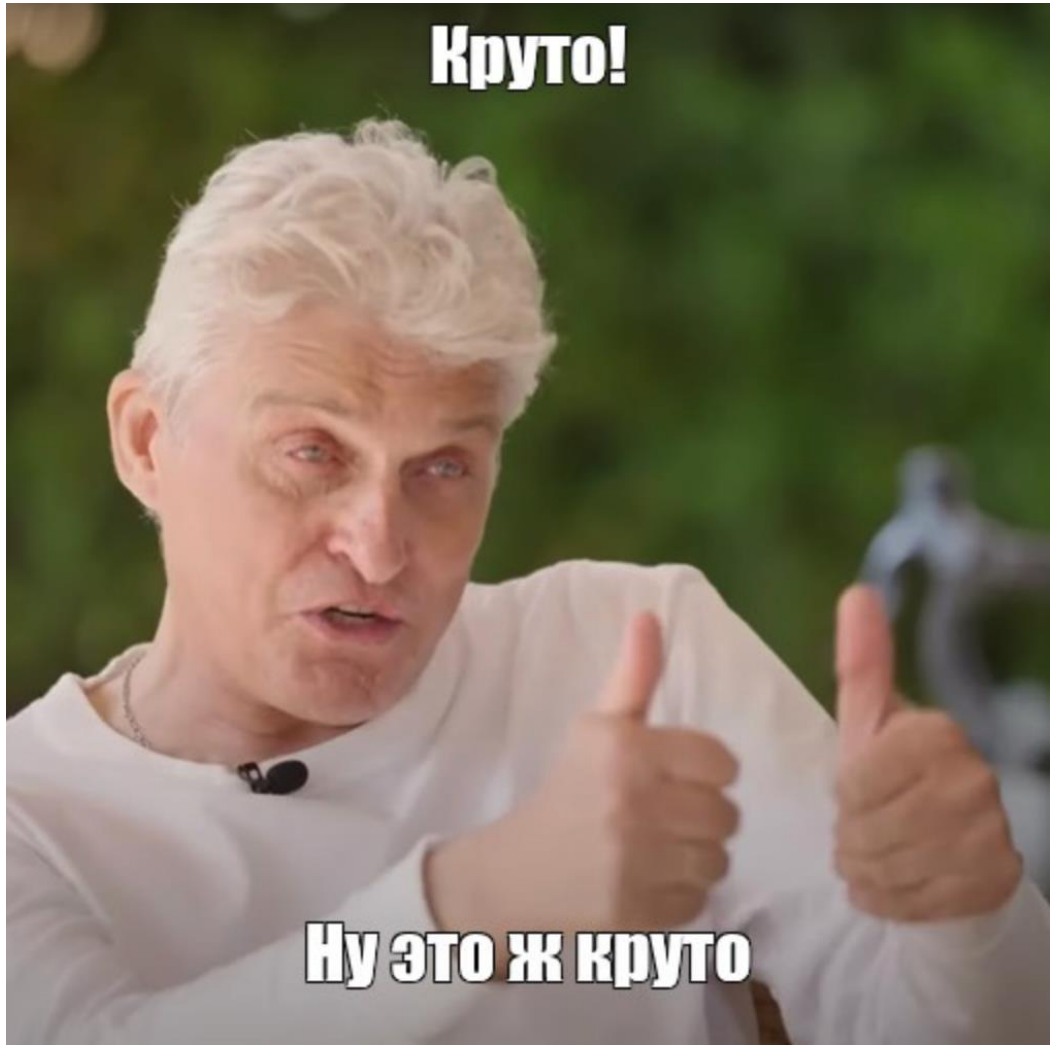
Exec.mem (JVM + Native)

50 Gb

25+15 Gb

4+20 Gb

Job 3 – Gold



«ДАННОЕ ЛИЦО ВКЛЮЧЕНО В РЕЕСТР ИНОСТРАННЫХ АГЕНТОВ МИНИСТЕРСТВА ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Job 4 – Gold


```
98
99   // files: 1 250 rows: 2 302 863 614 size: 185 GiB
100  val preAgg = spark.read.parquet("/foo/bar/pre_agg")
101
102  // files: 900 rows: 1 353 945 594 size: 89 GiB
103  val bigDimDf =  spark.read.parquet("/foo/bar/big_dimension")
104
105  val goldDf = preAgg
106    .groupBy()  // 50 simple columns
107    .agg(
108      sum($"sum_col").as("sum_col"),
109      sum($"cnt").as("cnt"),
110      max($"dttm").as("dttm"),
111      max($"ex").as("ex")
112    )
113    .transform()   //withColumn case when
114    .join(bigDimDf, "col1", "left")
115    .transform().   //withColumn case when
116    .withColumn("shard", pmod(hash(col(colName)), lit(N)) + 1)
117    .withColumn("ts_ins", current_timestamp())
118    .drop("col_69")
119
```

Job 4 – Gold

Exec.mem (JVM + Native)

No engine

6.5 мин

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	save at [redacted].scala:1030 save at [redacted].scala:1030	2025/09/16 09:31:00	5.3 min	5/5	8463/8463

Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (23)
+- WriteFiles (22)
   +- * Project (21)
      +- * Sort (20)
         +- * Project (19)
            +- Exchange (18)
```

34 Gb

Comet

11 мин

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	save at [redacted].scala:1030 save at [redacted].scala:1030	2025/09/16 09:31:42	6.6 min	5/5	10600/10600

Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (25)
+- WriteFiles (24)
   +- * Sort (23)
      +- Exchange (22)
         +- * Project (21)
            +- * CometColumnarToRow (20)
```

15+7 Gb

Velox

9,7 мин

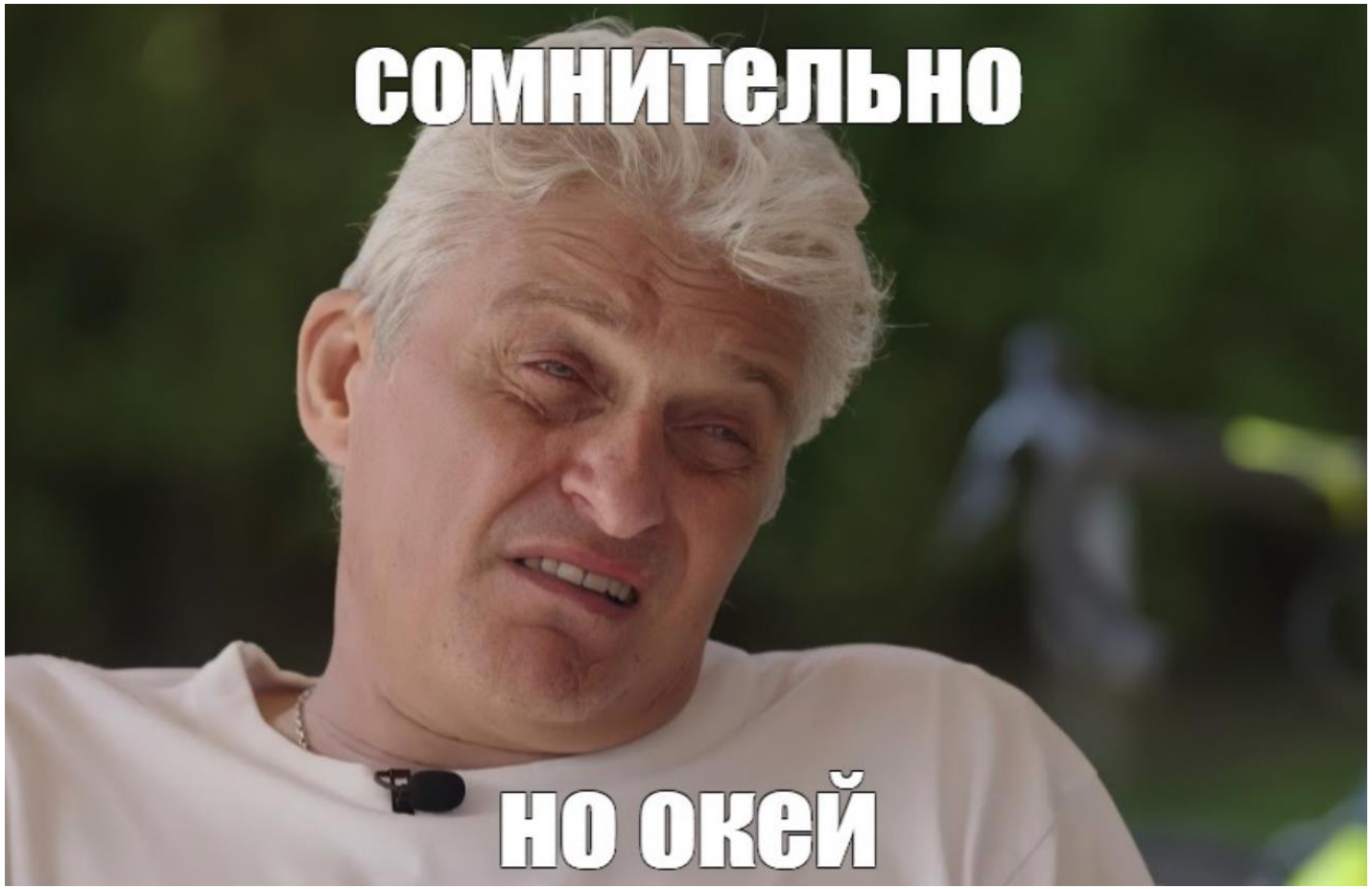
Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	save at [redacted].scala:1030 save at [redacted].scala:1030	2025/09/16 09:31:13	7.7 min	5/5	10600/10600 (1 failed)

Details

```
== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand (43)
+- VeloxColumnarWriteFiles (42)
   :- ^ WriteFilesExecTransformer (37)
   :  +- ^ InputIteratorTransformer (36)
   :     +- ColumnarExchange (34)
   :        +- VeloxResizeBatches (33)
   :           +- ^ ProjectExecTransformer (31)
```

4+19 Gb

Job 4 – Gold



«ДАННОЕ ЛИЦО ВКЛЮЧЕНО В РЕЕСТР ИНОСТРАННЫХ АГЕНТОВ МИНИСТЕРСТВА ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Выводы/советы

Что в итоге?

- Всего 12 job, полезных – 6.
- В основном – экономия RAM.
- Ускорения нет. 
- Gold – норм, silver – стрём.
- Fallback – зло
- Под движки джобы придётся переписать
- Сбойные ноды - зло
- Hash Join в Gluten и Push Based Shuffle в Comet – такоэ себе

Болячки

Gluten

spark.gluten.sql.columnar.forceShuffledHashJoin = false
spark.gluten.sql.columnar.sortMergeJoin = true
spark.sql.join.preferSortMergeJoin = true

```
Caused by: org.apache.gluten.exception.GlutenException: org.apache.gluten.exception.GlutenException: Exception: VeloxRuntimeError
Error Source: RUNTIME
Error Code: GENERIC_SPILL_FAILURE
Reason: Spill bytes will overflow. Bytes 10288278949, kMaxSpillBytesPerWrite: 2147483647
Retriable: True
Context: Operator: HashProbe[25] 22
Function: validateSpillBytesSize
File: /root/src/apache/incubator-gluten/ep/build-velox/build/velox_ep/velox/exec/Spill.cpp
Line: 137
Stack trace:
# 0  _ZN8facebook5velox7process10StackTraceC1Ei
# 1  _ZN8facebook5velox14VeloxExceptionC1EPKcmS3_St17basic_string_viewIcSt11char_traitsIcEES7_S7_S7_bNS1_4TypeES7_
# 2  _ZN8facebook5velox6detail14veloxCheckFailINS0_17VeloxRuntimeErrorERKSsEEvRKNS1_18VeloxCheckFailArgsET0_
# 3  _ZN8facebook5velox4exec10SpillState22validateSpillBytesSizeEm
```

APACHE DATAFUSION COMET™

spark.shuffle.push.enabled=false

Post scriptum

Post Scriptum

Попробуйте сами, приходите рассказать

Post Scriptum

Вопросы

СПАСИБО ЗА ВНИМАНИЕ!

Никита Благодарный
telegram @nblagodarnyy

ЧЕСТНЫЙ ЗНАК | центр развития перспективных технологий

SmartData 2025