

git_comments:

1. certain queries like `select count(*) from table` do not have any projected columns and still have `isReadAllColumns` as false in such cases `columnReaders` are not needed However, if `colsToInclude` is not empty we should initialize each `columnReader`
2. initialize the `rowbatchContext`

git_commits:

1. **summary:** HIVE-17874 : Parquet vectorization fails on tables with complex columns when there are no projected columns (Vihang Karajgaonkar, reviewed by Ferdinand Xu)
message: HIVE-17874 : Parquet vectorization fails on tables with complex columns when there are no projected columns (Vihang Karajgaonkar, reviewed by Ferdinand Xu)

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
label: code-design
2. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
3. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
label: code-design
4. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
5. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
6. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
7. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
label: test
8. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
9. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns

- description:** When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
10. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 11. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 12. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 13. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 14. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 15. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 16. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 17. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 18. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 19. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 20. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 21. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.
 22. **summary:** Parquet vectorization fails on tables with complex columns when there are no projected columns
description: When a parquet table contains an unsupported type like `{{Map}}`, `{{LIST}}` or `{{UNION}}` simple queries like `{{select count(*) from table}}` fails with `{{unsupported type exception}}` even though vectorized reader doesn't really need read the complex type into batches.

jira_issues_comments:

1. **body:** Thank you for the patch. Just a few minor comments. Is the last line of comments not needed or half done?
`{code:java} + //if there are colsToInclude initialize each columnReader {code}` I see the following is moving from constructor to the initial method. Is it just for clean up code? If so, not sure whether we can move `rbCtx = Utilities.getVectorizedRowBatchCtx(conf);` as well. `{code:java} colsToInclude = ColumnProjectionUtils.getReadColumnIDs(conf); {code}` Unnecessary change for the following line. `{code:java} + private VectorizedColumnReader buildVectorizedParquetReader({code}`

label: code-design

2. Here are the results of testing the latest attachment:

<https://issues.apache.org/jira/secure/attachment/12893472/HIVE-17874.01.patch> {color:green}SUCCESS:{color} +1 due to 1 test(s) being added or modified. {color:red}ERROR:{color} -1 due to 10 failed/errored test(s), 11317 tests executed *Failed tests:* {noformat}

org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver[auto_sortmerge_join_2] (batchId=47)
org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver[vectorization_parquet_projection] (batchId=42)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[llap_acid_fast] (batchId=156)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[optimize_nullscan] (batchId=163)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[resourceplan] (batchId=158)
org.apache.hadoop.hive.cli.TestMiniTezCliDriver.testCliDriver[explainanalyze_2] (batchId=101)
org.apache.hadoop.hive.cli.TestSparkCliDriver.testCliDriver[vectorization_parquet_projection] (batchId=121)
org.apache.hadoop.hive.cli.control.TestDanglingQOuts.checkDanglingQOut (batchId=204)
org.apache.hadoop.hive.ql.parse.TestReplicationScenarios.testConstraints (batchId=221)
org.apache.hadoop.hive.ql.parse.authorization.plugin.sqlstd.TestOperation2Privilege.checkHiveOperationTypeMatch (batchId=269) {noformat} Test results: <https://builds.apache.org/job/PreCommit-HIVE-Build/7439/testReport>
Console output: <https://builds.apache.org/job/PreCommit-HIVE-Build/7439/console> Test logs: <http://104.198.109.242/logs/PreCommit-HIVE-Build-7439/> Messages: {noformat} Executing
org.apache.hive.ptest.execution.TestCheckPhase Executing org.apache.hive.ptest.execution.PrePhase Executing
org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests
exited with: TestsFailedException: 10 tests failed {noformat} This message is automatically generated.
ATTACHMENT ID: 12893472 - PreCommit-HIVE-Build

3. **body:** Thanks for the review [~Ferd]. I made changes as you suggested. I moved `{{colsToInclude = ColumnProjectionUtils.getReadColumnIDs(conf);}}` in the `{{initialize}}` method because I got rid of unnecessary field `{{indexColumnsWanted}}` and reused `colsToInclude` instead. I have moved the `{{rbCtx = Utilities.getVectorizedRowBatchCtx(conf);}}` in the `initialize` method as well like you suggested. Also updated the comment and removed unnecessary diff. Feel free to let me know if you want me to publish the patch on RB as well.

label: code-design

4. LGTM +1 pending on the Precommit.

5. Here are the results of testing the latest attachment:

<https://issues.apache.org/jira/secure/attachment/12893482/HIVE-17874.02.patch> {color:green}SUCCESS:{color} +1 due to 1 test(s) being added or modified. {color:red}ERROR:{color} -1 due to 9 failed/errored test(s), 11317 tests executed *Failed tests:* {noformat}

org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver[vectorization_parquet_projection] (batchId=42)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[optimize_nullscan] (batchId=163)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[resourceplan] (batchId=158)
org.apache.hadoop.hive.cli.TestSparkCliDriver.testCliDriver[vectorization_parquet_projection] (batchId=121)
org.apache.hadoop.hive.cli.TestSparkPerfCliDriver.testCliDriver[query39] (batchId=243)
org.apache.hadoop.hive.cli.control.TestDanglingQOuts.checkDanglingQOut (batchId=204)
org.apache.hadoop.hive.ql.io.parquet.TestVectorizedColumnReader.testNullSplitForParquetReader (batchId=262)
org.apache.hadoop.hive.ql.parse.TestReplicationScenarios.testConstraints (batchId=221)
org.apache.hadoop.hive.ql.parse.authorization.plugin.sqlstd.TestOperation2Privilege.checkHiveOperationTypeMatch (batchId=269) {noformat} Test results: <https://builds.apache.org/job/PreCommit-HIVE-Build/7441/testReport>
Console output: <https://builds.apache.org/job/PreCommit-HIVE-Build/7441/console> Test logs: <http://104.198.109.242/logs/PreCommit-HIVE-Build-7441/> Messages: {noformat} Executing
org.apache.hive.ptest.execution.TestCheckPhase Executing org.apache.hive.ptest.execution.PrePhase Executing
org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests
exited with: TestsFailedException: 9 tests failed {noformat} This message is automatically generated.
ATTACHMENT ID: 12893482 - PreCommit-HIVE-Build

6. Hi [~vihangk1], can you help check the failed test cases?

7. **body:** The tests are working for me locally. I added `{{--SORT_QUERY_RESULTS}}` to make sure these are not flaky failures. Attaching the patch one more time.

label: test

8. Attaching it one more time to trigger precommit.

9. Here are the results of testing the latest attachment:

<https://issues.apache.org/jira/secure/attachment/12893758/HIVE-17874.04.patch> {color:green}SUCCESS:{color} +1 due to 1 test(s) being added or modified. {color:red}ERROR:{color} -1 due to 9 failed/errored test(s), 11324 tests executed *Failed tests:* {noformat} org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver[index_skewtable] (batchId=79) org.apache.hadoop.hive.cli.TestHBaseCliDriver.testCliDriver[hbase_queries] (batchId=96)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[llap_acid_fast] (batchId=156)
org.apache.hadoop.hive.cli.TestMiniTezCliDriver.testCliDriver[explainanalyze_2] (batchId=101)
org.apache.hadoop.hive.cli.TestSparkCliDriver.testCliDriver[subquery_multi] (batchId=110)
org.apache.hadoop.hive.cli.control.TestDanglingQOuts.checkDanglingQOut (batchId=205)

- org.apache.hadoop.hive ql.io.parquet.TestVectorizedColumnReader.testNullSplitForParquetReader (batchId=263)
org.apache.hadoop.hive ql.parse.TestReplicationScenarios.testConstraints (batchId=222)
org.apache.hadoop.hive ql.parse.authorization.plugin.sqlstd.TestOperation2Privilege.checkHiveOperationTypeMatch (batchId=270) {noformat} Test results: <https://builds.apache.org/job/PreCommit-HIVE-Build/7465/testReport>
Console output: <https://builds.apache.org/job/PreCommit-HIVE-Build/7465/console> Test logs:
<http://104.198.109.242/logs/PreCommit-HIVE-Build-7465/> Messages: {noformat} Executing
org.apache.hive.ptest.execution.TestCheckPhase Executing org.apache.hive.ptest.execution.PrePhase Executing
org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests
exited with: TestsFailedException: 9 tests failed {noformat} This message is automatically generated.
ATTACHMENT ID: 12893758 - PreCommit-HIVE-Build
10. Fixed TestVectorizedColumnReader test failure.
11. [~vihangk1], HIVE-14826 touches your code. Need to rebase your code on that.
12. Rebased and synced to the top. Attaching version 6 patch.
13. Here are the results of testing the latest attachment:
<https://issues.apache.org/jira/secure/attachment/12894158/HIVE-17874.06.patch> {color:green}SUCCESS:{color}
+1 due to 1 test(s) being added or modified. {color:red}ERROR:{color} -1 due to 6 failed/errored test(s), 11327
tests executed *Failed tests:* {noformat}
org.apache.hadoop.hive.cli.TestMiniLlapCliDriver.testCliDriver[orc_ppd_schema_evol_3a] (batchId=145)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[llap_acid] (batchId=164)
org.apache.hadoop.hive.cli.control.TestDanglingQOuts.checkDanglingQOut (batchId=205)
org.apache.hadoop.hive ql.parse.TestReplicationScenarios.testConstraints (batchId=222)
org.apache.hive.jdbc.TestTriggersTezSessionPoolManager.testTriggerHighShuffleBytes (batchId=229)
org.apache.hive.jdbc.TestTriggersWorkloadManager.testTriggerHighShuffleBytes (batchId=229) {noformat} Test
results: <https://builds.apache.org/job/PreCommit-HIVE-Build/7488/testReport> Console output:
<https://builds.apache.org/job/PreCommit-HIVE-Build/7488/console> Test logs:
<http://104.198.109.242/logs/PreCommit-HIVE-Build-7488/> Messages: {noformat} Executing
org.apache.hive.ptest.execution.TestCheckPhase Executing org.apache.hive.ptest.execution.PrePhase Executing
org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests
exited with: TestsFailedException: 6 tests failed {noformat} This message is automatically generated.
ATTACHMENT ID: 12894158 - PreCommit-HIVE-Build
14. test failures are unrelated. +1
15. Patch merged to master.
16. attaching branch-2 patch.
17. Reopening to trigger precommit on branch-2 patch.
18. Here are the results of testing the latest attachment:
<https://issues.apache.org/jira/secure/attachment/12895523/HIVE-17874.08-branch-2.patch>
{color:green}SUCCESS:{color} +1 due to 1 test(s) being added or modified. {color:red}ERROR:{color} -1 due to
10 failed/errored test(s), 10661 tests executed *Failed tests:* {noformat}
org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver[explaindependencydiffengs] (batchId=38)
org.apache.hadoop.hive.cli.TestMiniLlapCliDriver.testCliDriver[llap_smb] (batchId=142)
org.apache.hadoop.hive.cli.TestMiniLlapCliDriver.testCliDriver[orc_ppd_basic] (batchId=139)
org.apache.hadoop.hive.cli.TestMiniLlapCliDriver.testCliDriver[table_nonprintable] (batchId=140)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[join_acid_non_acid] (batchId=158)
org.apache.hadoop.hive.cli.TestMiniLlapLocalCliDriver.testCliDriver[union_fast_stats] (batchId=153)
org.apache.hadoop.hive.cli.TestNegativeCliDriver.testCliDriver[merge_negative_5] (batchId=88)
org.apache.hadoop.hive.cli.TestSparkCliDriver.testCliDriver[explaindependencydiffengs] (batchId=115)
org.apache.hadoop.hive.cli.TestSparkCliDriver.testCliDriver[vectorized_ptf] (batchId=125)
org.apache.hive.hcatalog.api.TestHCatClient.testTransportFailure (batchId=176) {noformat} Test results:
<https://builds.apache.org/job/PreCommit-HIVE-Build/7619/testReport> Console output:
<https://builds.apache.org/job/PreCommit-HIVE-Build/7619/console> Test logs:
<http://104.198.109.242/logs/PreCommit-HIVE-Build-7619/> Messages: {noformat} Executing
org.apache.hive.ptest.execution.TestCheckPhase Executing org.apache.hive.ptest.execution.PrePhase Executing
org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests
exited with: TestsFailedException: 10 tests failed {noformat} This message is automatically generated.
ATTACHMENT ID: 12895523 - PreCommit-HIVE-Build
19. Failures are unrelated. Committed to branch-2
20. This jira is resolved and released with Hive 3.0 If you find an issue with it, please create a new jira.