Item 104
git_comments:

git_commits:

1. **summary:** [SPARK-23097][SQL][SS] Migrate text socket source to V2
   message: [SPARK-23097][SQL][SS] Migrate text socket source to V2 ## What changes were proposed in this pull request? This PR moves structured streaming text socket source to V2. Questions: do we need to remove old "socket" source? ## How was this patch tested? Unit test and manual verification. Author: jerryshao <sshao@hortonworks.com> Closes #20382 from jerryshao/SPARK-23097.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-23097][SQL][SS] Migrate text socket source to V2
   body: ## What changes were proposed in this pull request? This PR moves structured streaming text socket source to V2. Questions: do we need to remove old "socket" source? ## How was this patch tested? Unit test and manual verification.

github_pulls_comments:

1. **[Test build #86581 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/86581/testReport)** for PR 20382 at commit [`8f3b548`]
   (https://github.com/apache/spark/commit/8f3b54824d92123a0e7d468d42db30dba72cded1). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
2. @jose-torres can you please help to review, thanks!
3. **[Test build #86640 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/86640/testReport)** for PR 20382 at commit [`56c60f3`]
   (https://github.com/apache/spark/commit/56c60f3d9d920cea095e78695544b371435ca6f5). * This patch **fails Spark unit tests**. * This patch merges cleanly. * This patch adds no public classes.
4. It's unfortunate that the socket tests don't actually run streams end to end, but I think that's orthogonal to this PR. Can you run one of the programming guide examples using socket source (e.g. org.apache.spark.examples.sql.streaming.StructuredSessionization) to make sure it works after this PR? If it does, LGTM
5. Jenkins, retest this please.
6. Hi @jose-torres , thanks for your reviewing. I tried both the example you mentioned and simple spark-shell command, I think it works, but the path will always go to V2 `MicroBatchReader` (still need you PR to fallback to V1 Source).
7. Right, that makes sense. LGTM
8. **[Test build #86671 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/86671/testReport)** for PR 20382 at commit [`56c60f3`]
   (https://github.com/apache/spark/commit/56c60f3d9d920cea095e78695544b371435ca6f5). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
9. **[Test build #86677 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/86677/testReport)** for PR 20382 at commit [`9ceb3be`]
   (https://github.com/apache/spark/commit/9ceb3be4a5a7a451ae740cf563792636f879a81b). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
10. @zsxwing @tdas would you please help to review, thanks!
11. I am holding off further comments on this PR until the major change of eliminating v1 Source is done. That would cause significant refactoring (including the fact that the common trait wont be needed). BTW, I strongly suggest moving the socket code to execution.streaming.sources, like other v2 sources.
12. Sure, will waiting for others to be merged, thanks @tdas .

13. #20445 will be merged in a few hours. please go ahead and update your PR with the refactoring that was suggested (mainly, no v1 version).
14. Sure, I will do it.
15. **[Test build #87199 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87199/testReport)** for PR 20382 at commit [`fdc9b9c`] (https://github.com/apache/spark/commit/fdc9b9c8a1dcc749be97cfd1c46a502c33bf4bb9). * This patch **fails due to an unknown error code, -9**. * This patch merges cleanly. * This patch adds the following public classes _(experimental)_: * `class TextSocketMicroBatchReader(options: DataSourceOptions) extends MicroBatchReader with Logging `
16. **[Test build #87202 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87202/testReport)** for PR 20382 at commit [`874c91c`] (https://github.com/apache/spark/commit/874c91c41942972cabb85be175f929fc62e74af7). * This patch **fails due to an unknown error code, -9**. * This patch merges cleanly. * This patch adds no public classes.
17. jenkins test this please
18. **[Test build #87203 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87203/testReport)** for PR 20382 at commit [`874c91c`] (https://github.com/apache/spark/commit/874c91c41942972cabb85be175f929fc62e74af7). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
19. Hi @tdas , would you please help to review again, thanks!
20. **[Test build #87371 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87371/testReport)** for PR 20382 at commit [`647c5cd`] (https://github.com/apache/spark/commit/647c5cdd1e3cb4138b597bd429e01308f50468a6). * This patch **fails to build**. * This patch merges cleanly. * This patch adds no public classes.
21. **[Test build #87372 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87372/testReport)** for PR 20382 at commit [`f3fc90c`] (https://github.com/apache/spark/commit/f3fc90cc94210f313861625b5a8fe6ef754c05bd). * This patch **fails due to an unknown error code, -9**. * This patch merges cleanly. * This patch adds no public classes.
22. **[Test build #87370 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87370/testReport)** for PR 20382 at commit [`068c050`] (https://github.com/apache/spark/commit/068c050547a3ae002ac77d0ea2d48e2b82caa049). * This patch **fails due to an unknown error code, -9**. * This patch **does not merge cleanly**. * This patch adds no public classes.
23. Jenkins, retest this please.
24. **[Test build #87383 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87383/testReport)** for PR 20382 at commit [`f3fc90c`] (https://github.com/apache/spark/commit/f3fc90cc94210f313861625b5a8fe6ef754c05bd). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
25. @jerryshao any updates?
26. Hi @tdas, I'm on vacation this week, will update the code when I have time. Sorry for the delay.
27. Aah okay. Thanks for letting me know.
28. @jerryshao ping on this.
29. Sorry @tdas for the delay. I'm working on this, will push new changes soon.
30. **[Test build #87659 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87659/testReport)** for PR 20382 at commit [`5011372`] (https://github.com/apache/spark/commit/501137269c983e4d028eba817d1c5f45a305171d). * This patch **fails Spark unit tests**. * This patch merges cleanly. * This patch adds no public classes.
31. **[Test build #87660 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87660/testReport)** for PR 20382 at commit [`fd890ad`]

(https://github.com/apache/spark/commit/fd890ad837bb7068c70a27921d67af1c3fe65350). * This patch **fails Spark unit tests**. * This patch merges cleanly. * This patch adds no public classes.
32. Jenkins, retest this please.
33. **[Test build #87664 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87664/testReport)** for PR 20382 at commit [`fd890ad`] (https://github.com/apache/spark/commit/fd890ad837bb7068c70a27921d67af1c3fe65350). * This patch **fails due to an unknown error code, -9**. * This patch merges cleanly. * This patch adds no public classes.
34. Jenkins, retest this please.
35. **[Test build #87667 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87667/testReport)** for PR 20382 at commit [`fd890ad`] (https://github.com/apache/spark/commit/fd890ad837bb7068c70a27921d67af1c3fe65350). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
36. @jerryshao please address the above comment, then we are good to merge!
37. Sure, I will do it today.
38. **[Test build #87819 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87819/testReport)** for PR 20382 at commit [`1073be4`] (https://github.com/apache/spark/commit/1073be420b2cc5fd099929fc0215bf8c1be4b6e0). * This patch **fails Spark unit tests**. * This patch merges cleanly. * This patch adds no public classes.
39. relevant test failed. please make sure that there is no flakiness in the tests.
40. **[Test build #87825 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87825/testReport)** for PR 20382 at commit [`6d38bed`] (https://github.com/apache/spark/commit/6d38bed38f01a6a3919e07587a00279ec4388f23). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
41. **[Test build #87831 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/87831/testReport)** for PR 20382 at commit [`762f1da`] (https://github.com/apache/spark/commit/762f1da952eae99fc7b377a08267c0d4cdaf00ee). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
42. LGTM. Merging to master.

github_pulls_reviews:

1. Change here to avoid triggering new distributed job.
2. The intent is for the V2 and V1 source to live in the same register, so existing queries can start using the V2 source with no change needed. This also allows the V2 implementation to be validated by passing all the old tests. RateSourceV2 is a bad example; it only exists because I didn't have time to write a fully compatible rate source. I'll work on fixing it.
3. To match the old parallelize behavior, the default number of partitions should be sparkContext.defaultParallelism.
4. Is it possible to initialize in the constructor?
5. nit: conversion to int is unnecessary
6. This is what I want to bring out. Originally I initialized this in constructor like old socket source. But I found that `MicroBatchReader` will be created in two different places with two objects. So initializing in constructor will create two sock threads and connectors. This is different from V1 source. In V1 source, we only created source once, but with V2 `MicroBatchReader` we will create two objects in two different places (one for schema), which means such side-affect actions in constructor will have two copies. Ideally we should only create this `MicroBatchReader` once.
7. I don't think this will solve that problem, since each reader will just have its own initialize bit. In general, I think it's fine if we do a bit of extra work. V1 sources do have to support being created multiple times (in e.g. restart scenarios), and the lifecycles of the two V2 readers being created here don't overlap. (We should be closing the tempReader created in DataStreamReader, though.)
8. @jose-torres , you mean that instead of creating a new V2 socket source, modifying current V1 socket source to make it work with V2, am I understanding correctly?

9. The idea is that the existing TextSocketSourceProvider will have the MicroBatchReadSupport implementation here, in addition to the StreamSourceProvider implementation it already has.
10. I see, thanks for the clarify. Let me change it.
11. this fix should go into 2.3 branch. thanks for catching this.
12. Why do we still need StreamSourceProvider?
13. If I don't misunderstand @jose-torres 's intention, basically he wanted this socket source to work also in V1 code path.
14. OK, I will create a separate PR for this small fix.
15. aah, i see earlier comments.
16. TD and I discussed this offline. It should be fine to remove the V1 StreamSourceProvider implementation, because: * this isn't a production-quality source, so users shouldn't need to fall back to it * this source won't be particularly useful at exercising the V1 execution pipeline once we transition all sources to V2
17. OK, I will update the patch accordingly.
18. Please add docs!! This is a base interface used by two source implementations. Also rename this such that its clear that this a base class and not an actual Reader (i.e. not a subclass of DataSourceV2 readers). Maybe `TextSocketReaderBase`
19. I would wait for my PR #20445 to go in where I migrate LongOffset to use OffsetV2
20. can you add a redirection in the `DataSource.backwardCompatibilityMap` for this?
21. nit: tutorials -> testing (i know it was like that, but lets fix it since we are changing it anyway)
22. this does not keep it forever. so remove this reason, just keep "no support for fault recover".
23. supernit: is there need for a variable here?
24. nit: wont this fit on a single line?
25. This shows up in the StreamingQueryProgressEvent as description, so it may be better to have it as "TextSocket[..."
26. why not check it as DataSourceOptions (which is known to be case-insensitive) rather than a map which raises questions about case sensitivity?
27. nit: side-affect -> side-effect. good catch.
28. i feel like this needs a try finally approach as well.
29. why does this show up as a new file? was this not a "git mv"? something went wrong, i would prefer that i can see a simple diff. Not much should change in the tests.
30. Tutorials is correct here; see e.g. StructuredSessionization.scala
31. Sorry @tdas , I did it by simply "mv", not "git mv". This doesn't change a lot, just to be suited for data source v2 API.
32. what happened to the input row metrics test?
33. These updated tests are getting more complicated with the direct calling of low-level data source APIs. Can you convert these tests to the more highlevel tests like Kafka? Well if it gets too complicated to make it work with `testStream` then you can simply use `query.processAllAvailable`. Then we wont have to worry about changing APIs any more.
34. please add a test for this!
35. Because of the changes of data source APIs, so for now it is hard to get DF's metrics from V2 API, that's why I deleted this test.
36. the test below seems good replacement to me.
37. there is a slim chance that batch2stamp will be same as batch1stamp. maybe worth adding a sleep(10) to ensure this. you should also check batch1stamp with timestamp taken directly before the query. otherwise it may pass tests if the query generated batch1stamp = -1 and batch2stamp = -2.
38. assert on the message.
39. this does not make sense. you are directly accessing something that should be accessed while synchronized on this.
40. Hi @tdas , what's the meaning of "you should also check batch1stamp with timestamp taken directly before the query. ", I'm not clearly sure what specifically are you pointing to?
41. In my local test, the assert message is `Can't assign requested address`, but on Jenkins, it is `Connection refused`. The difference might be due to different OS/native method. I think it would be better to not check the message due to different outputs. Even if we change to follow Jenkins way, it still fails in my local Mac.
42. val timestamp = System.currentTimeMillis testStream(...)(// get batch1stamp) // assert batch1stamp >= timestamp
43. I see. Will update it.
44. thats fine.

jira_issues:

jira_issues_comments: