Item 99
git_comments:

git_commits:

1. **summary:** [SPARK-4790][STREAMING] Fix ReceivedBlockTrackerSuite waits for old file...
   message: [SPARK-4790][STREAMING] Fix ReceivedBlockTrackerSuite waits for old file... ...s to get deleted before continuing. Since the deletes are happening asynchronously, the getFileStatus call might throw an exception in older HDFS versions, if the delete happens between the time listFiles is called on the directory and getFileStatus is called on the file in the getFileStatus method. This PR addresses this by adding an option to delete the files synchronously and then waiting for the deletion to complete before proceeding. Author: Hari Shreedharan <hshreedharan@apache.org> Closes #3726 from harishreedharan/spark-4790 and squashes the following commits: bbbacd1 [Hari Shreedharan] Call cleanUpOldLogs only once in the tests. 3255f17 [Hari Shreedharan] Add test for async deletion. Remove method from ReceiverTracker that does not take waitForCompletion. e4c83ec [Hari Shreedharan] Making waitForCompletion a mandatory param. Remove eventually from WALSuite since the cleanup method returns only after all files are deleted. af00fd1 [Hari Shreedharan] [SPARK-4790][STREAMING] Fix ReceivedBlockTrackerSuite waits for old files to get deleted before continuing.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-4790][STREAMING] Fix ReceivedBlockTrackerSuite waits for old file...
   body: ...s to get deleted before continuing. Since the deletes are happening asynchronously, the getFileStatus call might throw an exception in older HDFS versions, if the delete happens between the time listFiles is called on the directory and getFileStatus is called on the file in the getFileStatus method. This PR addresses this by adding an option to delete the files synchronously and then waiting for the deletion to complete before proceeding.

github_pulls_comments:

1. [Test build #24556 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24556/consoleFull) for PR 3726 at commit [`af00fd1`] (https://github.com/apache/spark/commit/af00fd145f05cb7bee2474ee37f366e1fd56912d). - This patch merges cleanly.
2. [Test build #24556 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24556/consoleFull) for PR 3726 at commit [`af00fd1`] (https://github.com/apache/spark/commit/af00fd145f05cb7bee2474ee37f366e1fd56912d). - This patch **fails Spark unit tests**. - This patch merges cleanly. - This patch adds the following public classes _(experimental)_: - `class HiveThriftServer2Listener(val server: HiveServer2) extends SparkListener`
3. Test FAILed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24556/ Test FAILed.
4. Failures are in the parquet suite - I don't think it is related to this PR.
5. Jenkins, retest this please
6. [Test build #24560 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24560/consoleFull) for PR 3726 at commit [`af00fd1`] (https://github.com/apache/spark/commit/af00fd145f05cb7bee2474ee37f366e1fd56912d). - This patch merges cleanly.
7. [Test build #24560 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24560/consoleFull) for PR 3726 at commit [`af00fd1`] (https://github.com/apache/spark/commit/af00fd145f05cb7bee2474ee37f366e1fd56912d). - This patch

fails Spark unit tests. - This patch merges cleanly. - This patch adds the following public classes _(experimental)_: - `class HiveThriftServer2Listener(val server: HiveServer2) extends SparkListener`

8. Test FAILed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24560/ Test FAILed.
9. Jenkins, retest this please.
10. [Test build #24562 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24562/consoleFull) for PR 3726 at commit [`af00fd1`] (https://github.com/apache/spark/commit/af00fd145f05cb7bee2474ee37f366e1fd56912d). - This patch merges cleanly.
11. [Test build #24562 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24562/consoleFull) for PR 3726 at commit [`af00fd1`] (https://github.com/apache/spark/commit/af00fd145f05cb7bee2474ee37f366e1fd56912d). - This patch **passes all tests**. - This patch merges cleanly. - This patch adds no public classes.
12. Test PASSed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24562/ Test PASSed.
13. Rather than changing the code to accommodate the tests, i think its better to tweak the test to make it account for the laziness. I would simply change the test [here] (https://github.com/apache/spark/pull/3726/files#diff-f623a1fd0c6039bd6eb1746f9cc692c5L173) by making the test in a `eventually` block.
14. > Rather than changing the code to accommodate the tests, i think its better to tweak the test to make it account for the laziness. I would simply change the test here by making the test in a eventually block. In #3721, I used a `eventually` block previously. It was override by the current fix with `Future`.
15. Eventually is pretty much trying a test in a loop and that encourages flakey tests - I am in general not a very big fan of using eventually. The correct approach is to fix the non-determinism. Being able to detect that all files have been deleted has value - and gives easy debuggability. Being able to deterministically to know that all files are gone before proceeding does help debuggability. I don't really believe we are actually a accommodating the test - we are really fixing the code to allow deterministic testing.
16. [Test build #24791 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24791/consoleFull) for PR 3726 at commit [`e4c83ec`] (https://github.com/apache/spark/commit/e4c83eca17c660644216b6d8cc862f8b1654649c). - This patch merges cleanly.
17. [Test build #24791 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24791/consoleFull) for PR 3726 at commit [`e4c83ec`] (https://github.com/apache/spark/commit/e4c83eca17c660644216b6d8cc862f8b1654649c). - This patch **fails PySpark unit tests**. - This patch merges cleanly. - This patch adds no public classes.
18. Test FAILed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24791/ Test FAILed.
19. Jenkins, retest this please
20. [Test build #24806 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24806/consoleFull) for PR 3726 at commit [`e4c83ec`] (https://github.com/apache/spark/commit/e4c83eca17c660644216b6d8cc862f8b1654649c). - This patch merges cleanly.
21. [Test build #24806 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24806/consoleFull) for PR 3726 at commit [`e4c83ec`] (https://github.com/apache/spark/commit/e4c83eca17c660644216b6d8cc862f8b1654649c). - This patch **passes all tests**. - This patch merges cleanly. - This patch adds no public classes.
22. Test PASSed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24806/ Test PASSed.
23. [Test build #24807 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24807/consoleFull) for PR 3726 at commit [`3255f17`] (https://github.com/apache/spark/commit/3255f1710d7ffbbbd2de4b098eb444885b2aad36). - This patch merges cleanly.

24. [Test build #24807 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24807/consoleFull) for PR 3726 at commit [`3255f17`] (https://github.com/apache/spark/commit/3255f1710d7ffbbbd2de4b098eb444885b2aad36). - This patch **fails Spark unit tests**. - This patch merges cleanly. - This patch adds no public classes.
25. Test FAILed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24807/ Test FAILed.
26. Jenkins, retest this please
27. [Test build #24811 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24811/consoleFull) for PR 3726 at commit [`3255f17`] (https://github.com/apache/spark/commit/3255f1710d7ffbbbd2de4b098eb444885b2aad36). - This patch merges cleanly.
28. [Test build #24811 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24811/consoleFull) for PR 3726 at commit [`3255f17`] (https://github.com/apache/spark/commit/3255f1710d7ffbbbd2de4b098eb444885b2aad36). - This patch **passes all tests**. - This patch merges cleanly. - This patch adds no public classes.
29. Test PASSed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24811/ Test PASSed.
30. Looking good, except one comment in the testsuite (and another optional comment) .
31. [Test build #24869 has started] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24869/consoleFull) for PR 3726 at commit [`bbbacd1`] (https://github.com/apache/spark/commit/bbbacd1a441e43ce46e49bea6c85c6d7834c5487). - This patch merges cleanly.
32. [Test build #24869 has finished] (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/24869/consoleFull) for PR 3726 at commit [`bbbacd1`] (https://github.com/apache/spark/commit/bbbacd1a441e43ce46e49bea6c85c6d7834c5487). - This patch **passes all tests**. - This patch merges cleanly. - This patch adds no public classes.
33. Test PASSed. Refer to this link for build results (access rights to CI server needed): https://amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/24869/ Test PASSed.
34. Hey @tdas, is this good to go now? It would be nice to eliminate this cause of test flakiness, since this seems to be causing a decent number of failures today.
35. Yeah, LGTM. Merging this.

github_pulls_reviews:

1. `1 second` may be not enough. Why not return Future to the user so that they can use it to wait by themselves?
2. I fixed it using `Future` in #3721. Could you take a look?
3. These two signatures can be merged into a single method with default parameters. In fact it is probably okay to make the second parameter non-optional, which would take care of @zsxwing valid concern of the method not being obvious that it does stuff asynchronously.
4. I think its better to not expose implementation details like asynchronous-ness in the signature (i.e. returning `Future`) unless absolutely necessary. Also, the timeout is only for testing, so this is not for actual usage. So its okay to have 1 second. @harishreedharan Could you document that in the scala doc of the method that the 2nd parameter = true is only for testing?
5. Yep, sound good. I will update the PR.
6. Why do we need two methods still? We can just have this version, and update the usages everywhere.
7. Can you extend this unit test to test for `waitForComplete = false` as well. Basically call `manager.cleanupOldLogs(manualClock.currentTime, waitForCompletion = false)` and see with `eventually` that more files gets deleted.
8. Its worth testing both code paths because the async code path is what is going to be used in production.
9. Hmm..I thought I pushed the commit that removed this one. Let me check. Anyway I agree we don't need both, will remove the other one.
10. This call should be made only once (as in real use), instead of being called multiple times from within a `eventually` block.

11. Actually, mind renaming this method to `cleanupOldBlocks`? Realized that it was inconsistent with `cleanupOldBatches` and `cleanupOldLogs`. I know its not your code :)

jira_issues:

jira_issues_comments: