HBASE-19876

chia7712

2018-01-28

Contents

Chapter 1

Root issue HBASE-19876

1.1 Summary

The exception happening in converting pb mutation to hbase.mutation messes up the CellScanner

1.2 Description

```
2018-01-27 22:51:43,794 INFO  [hconnection-0x3291b443-shared-pool11-t6] client.AsyncRequestFutureImpl(778): id=5, table=testQuotaStatusFromMaster3, attempt=6/16 failed=20ops, last exception=org.apache.hadoop.hbase.client.WrongRowIOException: org.apache.hadoop.hbase.client.WrongRowIOException: The row in xxx doesn't match the original one aaa
    at org.apache.hadoop.hbase.client.Mutation.add(Mutation.java:776)
    at org.apache.hadoop.hbase.client.Put.add(Put.java:282)
    at org.apache.hadoop.hbase.shaded.protobuf.ProtobufUtil.toPut(ProtobufUtil.java:642)
    at org.apache.hadoop.hbase.regionserver.RSRpcServices.doBatchOp(RSRpcServices.java:952)
    at org.apache.hadoop.hbase.regionserver.RSRpcServices.doNonAtomicRegionMutation(RSRpcServices.java:896)
    at org.apache.hadoop.hbase.regionserver.RSRpcServices.multi(RSRpcServices.java:2591)
    at org.apache.hadoop.hbase.shaded.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:41560)
    at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:404)
    at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:130)
    at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:324)
    at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:304)
```

I noticed this bug when testing the table space quota.

When rs are converting pb mutation to hbase.mutation, the quota exception or cell exception may be thrown.

```
Unable to find source-code formatter for language: rsrpcservices#dobatchop.java. Available languages are: actionscript, ada, applescript, bash, c, c#, c++, cpp, css, erlang, go, groovy, haskell, html, java, javascript, js, json, lua, none, nyan, objc, perl, php, python, r, rainbow, ruby, scala, sh, sql, swift, visualbasic, xml, yaml      for (ClientProtos.Action action: mutations) {
        MutationProto m = action.getMutation();
        Mutation mutation;
        if (m.getMutateType() == MutationType.PUT) {
          mutation = ProtobufUtil.toPut(m, cells);
          batchContainsPuts = true;
        } else {
          mutation = ProtobufUtil.toDelete(m, cells);
          batchContainsDelete = true;
        }
        mutationActionMap.put(mutation, action);
        mArray[i++] = mutation;
        checkCellSizeLimit(region, mutation);
        // Check if a space quota disallows this mutation
        spaceQuotaEnforcement.getPolicyEnforcement(region).check(mutation);
        quota.addMutation(mutation);
      }
```

rs has caught the exception but it doesn't have the cellscanner skip the failed cells.

```
    } catch (IOException ie) {
      if (atomic) {
        throw ie;
      }
      for (Action mutation : mutations) {
        builder.addResultOrException(getResultOrException(ie, mutation.getIndex()));
      }
    }
```

The bug results in the WrongRowIOException to remaining mutations since they refer to invalid cells.

1.3 Attachments

1. HBASE-19876.branch-1.2.v0.patch
2. HBASE-19876.master.001.patch
3. HBASE-19876.v0.patch
4. HBASE-19876.v1.patch
5. HBASE-19876.v2.patch
6. HBASE-19876.v3.patch
7. HBASE-19876.v3.patch
8. HBASE-19876.v3.patch
9. HBASE-19876.v3.patch
10. HBASE-19876.v4.patch
11. HBASE-19876.v5.patch
12. HBASE-19876.v6.patch

1.4 Comments

1. **anoop.hbase:** A related issue was fixed some time back.. Skipping the cur cell which is problematic. So that might be missing for some places exception handling !!!

2. **chia7712:** {quote}A related issue was fixed some time back.

   {quote}

   HBASE-15214

   {quote}Skipping the cur cell which is problematic. So that might be missing for some places exception handling !!!

   {quote}

   yep, all append/increment/doBatchOphave similar issue...

3. **chia7712:** The append/increment wont do the batch so only batch of put/Delete has this issue. That makes the fix easy. Will come back soon.

4. **anoop.hbase:** Append and increment Mutation may have N Cells in that (for same row). Even if one cell is malformed, we have to fail the op. Ya that make sense. Need to verify once though. Thanks Chia.

5. **yuzhihong@gmail.com:** Looks good overall

   ```

   1008            // if the mArray[i] != null, it means the cell scanner has be read.
   ```

   Typo: 'be'

   Please remove commented out code in the test.

6. **chia7712:** v1 address [~yuzhihong@gmail.com]'s comment

7. **chia7712:** The failed tests are caused by new timeout rule.HBASE-19911 is dealing with them.

   v2 fix the checkstyle warnings.

8. **chia7712:** v3 fix the failed test...

9. **stack:** .001 rebase of [~chia7712] patch.

10. **chia7712:** Thanks [~stack]. I forget to say this issue is blocked byHBASE-19900. IfHBASE-19900 is not resolved, the tests in the patch are unstable because of the region-level exception.

11. **chia7712:** Trying to write the test for branch-1.2 and branch-1.3...

12. **chia7712:** The failed tests pass locally...Try the patch for master. Ping for reviews also.

13. **stack:** Took a look. This is tricky stuff. Thanks for diving in.

    This comment is a little confusing:

    890 // TODO doBatchOp has handled the IOE for all non-atomic operations

    891 // Catching IOE here may confuse readers in the future

    It is in a catch for an IOE so ... should we remove the catch block this comment is in?

    I was trying to see if you could get rid of the cycle looking for Gets in checkAndRowMutate now it has been added to doBatchOp but couldn't figure it; an optimization for another time (smile).

    On below, I agree.

    1011 // The atomic ops use the global exception although I feel it is ok to add the exception

    1012 // to each action.

    We need an issue for the below?

    184 * The invalid cells is in rm. The rm should fail but the subsequent mutations should succeed.

    185 * Currently, we have no client api to submit the request consisting of condition-rm and mutation.

    186 * Hence, this test build the request manually.

    We need to make it so it is easy to stand up a RegionServer w/o a Master so you don't have to do the below. This should be easier now... since HBASE-18846. Let me work on it....

218 ClientProtos.MultiResponse response = rs.getRSRpcServices().multi(controller, request);

Great tests [~chia7712]

This patch is super important stuff.

+1 to commit.

14. **chia7712:** {quote}It is in a catch for an IOE so ... should we remove the catch block this comment is in?

    {quote}

    The non-atomicdoBatchOp won't throw IOE....The catch here is weird but it won't be used. We can do a minor refactor for it later.:)

    {quote}I was trying to see if you could get rid of the cycle looking for Gets in checkAndRowMutate now it has been added to doBatchOp but couldn't figure it; an optimization for another time (smile).

    {quote}

    ya, another refactor....:(

    {quote}On below, I agree.

    1011 // The atomic ops use the global exception although I feel it is ok to add the exception

    1012 // to each action.

    {quote}

    Oh...Iforgot to remove the comment. After HBASE-19900, our client can handle both of region exception and action exception accurately. Only adding the region exception can reduce the memoryconsumption (it is similar toHBASE-19715).

    {quote}We need an issue for the below?

    {quote}

    Agreed. It would be a nice feature for 2.0.

    {quote}We need to make it so it is easy to stand up a RegionServer w/o a Master so you don't have to do the below.

{quote}

The test is necessary for both of branch-1.3 and branch-1.2 as we won't add the such client APIs to them I guess. By contrast, if the impl can be introduced to client APIs for branch-1.4+, we cansubstitute the real use case for the hacky test.

{quote}+1 to commit.

{quote}

I'm onmytravels so I will commit the patchon Monday. Thanks for the reviews [~stack]

15. **stack:** bq. The non-atomic doBatchOp won't throw IOE....The catch here is weird but it won't be used. We can do a minor refactor for it later.

    Agree. On commit add in the above line?

    Otherwise all good. Makes sense that the test be portable but I should make it so standing up a RegionServer is easy in test; will cut down test times.

    +1 on commit Monday.

16. **chia7712:** {quote}On commit add in the above line?

    {quote}

    roger that

17. **chia7712:** Push this patch to all active branches. Thanks for all reviews.

18. **hudson:** SUCCESS: Integrated in Jenkins build HBase-1.3-IT #348 (See [https://builds.apache.org/job/HBase-1.3-IT/348/])
    HBASE-19876 The exception happening in converting pb mutation to (chia7712: rev e36dca55bcd5ad95cd99a19a363
    * (add) hbase-server/src/test/java/org/apache/hadoop/hbase/client/TestMalformedCellFromClient.java
    * (edit) hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/RSRpcServices.java
    * (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/RequestConverter.java

19. **hudson:** SUCCESS: Integrated in Jenkins build HBase-1.2-IT #1068 (See [https://builds.apache.org/job/HBase-1.2-IT/1068/])
    HBASE-19876 The exception happening in converting pb mutation to (chia7712: rev b59c3151dfe41926d242c5c581a2
    * (add) hbase-server/src/test/java/org/apache/hadoop/hbase/client/TestMalformedCellFromClient.java
    * (edit) hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/RSRpcServices.java
    * (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/RequestConverter.java

20. **hudson:** FAILURE: Integrated in Jenkins build HBase-Trunk_matrix #4579 (See [https://builds.apache.org/job/H
    Trunk_matrix/4579/])
    HBASE-19876 The exception happening in converting pb mutation to (chia7712: rev 2f48fdbb26ff555485b4aa3393d8

* (edit) hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/RSRpcServices.java
* (edit) hbase-server/src/test/java/org/apache/hadoop/hbase/client/TestMalformedCellFromClient.java
* (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/shaded/protobuf/RequestConverter.java

Chapter 2

Connected issue HBASE-19900

2.1 Summary

Region-level exception destroy the result of batch

2.2 Description

1) decrease action count repeatedly

If the AsyncRequestFuture#waitUntilDone return prematurely, user will get the incorrect results. Or user will be block by AsyncRequestFuture#waitUntilDone as the count is never equal with 0.

2) the successive result will be overwrited

3) The failed op is added to RetriesExhaustedWithDetailsException repeatedly

AsyncRequestFutureImpl#receiveMultiAction process the action-lever error first, and then add the region-level exception to each action. Hence, user may get the various exceptions for the same action (row op) from the RetriesExhaustedWithDetailsException.

In fact, if both of action-level exception and region-lever exception exist, they always have the same context. I'm not sure whether that is what RetriesExhaustedWithDetailsException want. As i see it, we shouldn't have the duplicate ops in RetriesExhaustedWithDetailsException since that may confuse users if they catch the RetriesExhaustedWithDetailsException to check the invalid operations.

2.3 Attachments

1. HBASE-19900.branch-1.4.v0.patch
2. HBASE-19900.v0.patch
3. HBASE-19900.v1.patch
4. HBASE-19900.v2.patch
5. HBASE-19900.v2.patch
6. HBASE-19900.v2.patch

2.4 Comments

1. **chia7712:** The patch merge the check of action-level exception and region-lever exception.If the action fail without any exception, we assign the region-level exception to it. If no region-level exception exists, we assign a unknown errorinstead. In short, each action won't be handled repeatedly so the following issues should be fixed.

   # decrease action count repeatedly

   # the successive result will be overwrited

   # The failed op is added to RetriesExhaustedWithDetailsException repeatedly

   The patch also add some debuginfo to trace theoverwrite of result and exception.

2. **chia7712:** All bugs can be easily reproduced by batching the RowMutations. However, only 1.4+ support to do the batch ofRowMutations (HBASE-19096). I am not sure whether any real use case can be used to reproduce the bug for branch-1.2 and branch-1.3.

3. **chia7712:** Seems some tests expect thatfailed ops inRetriesExhaustedWithDetailsException will have the different exceptions.

4. **chia7712:** TestDLSFSHLog is traced byHBASE-19554. Ping for reviews. If no objectionagainst the approach, I will prepare the patch for branch-1.x

5. **yuzhihong@gmail.com:** Looks good overall.

   ```

   +    Retry canRetry = null;
   ```

   You changed the variable type. Variable name should be changed to reflect this.

   cleanServerCache() calls clearCaches(). Better align the method name to be consistent.

   ```

   +  long getActionsInProgress() {
   ```

   Actions are not returned, only the count. getActionsInProgress -> getNumberOfActionsInProgress

6. **stack:** Skimmed. Looks excellent. +1 Must have been a pain to track down.

7. **chia7712:** v2 patch address [~yuzhihong@gmail.com]'s comment.

8. **yuzhihong@gmail.com:** +1

9. **chia7712:** retry. The failed tests pass locally.

10. **chia7712:** Will commit the fix to all active branches.

11. **chia7712:** Thanks for all reviews.

12. **hudson:** SUCCESS: Integrated in Jenkins build HBase-1.3-IT #347 (See [https://builds.apache.org/job/HBase-1.3-IT/347/])
    HBASE-19900 Region-level exception destroy the result of batch (chia7712: rev 32b5f514b1a309375bd51d968ca19ef6
    * (add) hbase-client/src/test/java/org/apache/hadoop/hbase/client/TestAsyncProcessWithRegionException.java
    * (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/client/AsyncProcess.java

13. **hudson:** SUCCESS: Integrated in Jenkins build HBase-1.2-IT #1067 (See [https://builds.apache.org/job/HBase-1.2-IT/1067/])
    HBASE-19900 Region-level exception destroy the result of batch (chia7712: rev ebdfbaf2a090dde5685944604f61a8b3
    * (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/client/AsyncProcess.java
    * (add) hbase-client/src/test/java/org/apache/hadoop/hbase/client/TestAsyncProcessWithRegionException.java

14. **hudson:** FAILURE: Integrated in Jenkins build HBase-Trunk_matrix #4540 (See [https://builds.apache.org/job/H
    Trunk_matrix/4540/])
    HBASE-19900 Region-level exception destroy the result of batch (chia7712: rev d8b999e6950bc01e8aab8ecce437e710
    * (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/client/AsyncRequestFutureImpl.java
    * (add) hbase-server/src/test/java/org/apache/hadoop/hbase/client/TestMalformedCellFromClient.java
    * (edit) hbase-client/src/main/java/org/apache/hadoop/hbase/client/AsyncBatchRpcRetryingCaller.java
    * (add) hbase-client/src/test/java/org/apache/hadoop/hbase/client/TestAsyncProcessWithRegionException.java

15. **xvxinhui:** wonder why you clear the whole cache for a server via invoking cleanServerCache() in this revision, isn't it too overreacted, any cornor case related? [~chia7712]