

HBASE-22917

pankaj2461

2019-08-25

# Contents

# Chapter 1

## Root issue HBASE-22917

### 1.1 Summary

Proc-WAL roll fails always saying someone else has already created log

### 1.2 Description

Recently we met a weird scenario where Procedure WAL roll fails as it is already created by someone else.

Later while going through the logs and code, observed that during Proc-WAL roll it failed to write the header. On failure file stream is just closed,

```
1  try {  
2  ProcedureWALFormat.writeHeader(newStream, header);  
3  startPos = newStream.getPos();  
4  } catch (IOException ioe) {  
5  LOG.warn("Encountered exception writing header", ioe);  
6  newStream.close();  
7  return false;  
8  }
```

Since we don't delete the corrupted file or increment the `*flushLogId*`, so on each retry it is trying to create the same `*flushLogId*` file. However Hmaster failover will resolve this issue, but we should handle it.

### 1.3 Attachments

No attachments

### 1.4 Comments

1. **stack:** Merged. Thanks for patch [~pankaj2461]
2. **zhangduo:** I do not think it is fine to merge this back.

This is the feedback on the PR

{quote}

IIRC two master can't write proc-WAL together, do we have such corner scenario?

However here we are trying to cleanup when rollWriter fails when write header throws IOE.

{quote}

Actually it is our magic on the file id which prevents two masters write the proc-WAL together, so we should not use this as a assumption to implement our file id logic, totally wrong.

Now we just increase the file id by one if we failed to delete the old file, but this is an rpc call right? It could happen that on the NN side, the file has been deleted successfully but at client side we get an error, and then we increase the file id by 1, and then there will be a whole, what if another master tries to write new file id but just fill in the whole? Then we have two 'live' masters which could both write proc wal(at least there be a small overlap due to the aysnc behavior on zk session expire processing). This will lead to inconsistency and mess up everything.

So my suggestion is that, unless we have a clear explanation that the above scenario can not happen, then the safest way is to just abort the HMaster if we fail to roll the writer. And maybe it is safe to just increase the file id without deleting the broken proc wal file(this is a typical solution in WAL based system), but anyway, usually deleting a wal file is not a good idea...

Thanks.

3. **hudson:** Results for branch master

[build #1522 on builds.a.o|<https://builds.apache.org/job/HBase%20Nightly/job/master/1522/>]:

(x) \*{color:red}-1 overall{color}\*  
----

details (if available):

(/) {color:green}+1 general checks{color}

-- For more information [see general report|<https://builds.apache.org/job/HBase%20Nightly/job/master/1522//General>]

(/) {color:green}+1 jdk8 hadoop2 checks{color}  
-- For more information [see jdk8 (hadoop2) report|https://builds.apache.org/job/HBase%20Nightly/job/master/15]

(x) {color:red}-1 jdk8 hadoop3 checks{color}  
-- Something went wrong running this stage, please [check relevant console output|https://builds.apache.org/job/HB

(/) {color:green}+1 source release artifact{color}  
-- See build output for details.

(/) {color:green}+1 client integration test{color}

4. **stack:** Reasonable. Let me revert.

5. **stack:** Reverted from branch-2.2+. The PR has been closed as merged. Make a new one [~pankaj2461] after we have answers for [~zhangduo] query above? Thanks sir.

6. **hudson:** Results for branch branch-2.2  
[build #679 on builds.a.o|https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/679/]:  
(x) \*{color:red}-1 overall{color}\*  
----

details (if available):

(/) {color:green}+1 general checks{color}  
-- For more information [see general report|https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/679//General\_Nightly\_Build\_Report/]

(/) {color:green}+1 jdk8 hadoop2 checks{color}  
-- For more information [see jdk8 (hadoop2) report|https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/679//JDK8\_Nightly\_Build\_Report\_(Hadoop2)/]

(x) {color:red}-1 jdk8 hadoop3 checks{color}  
-- For more information [see jdk8 (hadoop3) report|https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/679//JDK8\_Nightly\_Build\_Report\_(Hadoop3)/]

(/) {color:green}+1 source release artifact{color}  
-- See build output for details.

(/) {color:green}+1 client integration test{color}

7. **hudson:** Results for branch branch-2  
[build #2340 on builds.a.o|https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340/]:  
(x) \*{color:red}-1 overall{color}\*  
----

details (if available):

(/) {color:green}+1 general checks{color}  
-- For more information [see general report|[https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340//General\\_Nightly\\_Build\\_Report/](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340//General_Nightly_Build_Report/)]

(x) {color:red}-1 jdk8 hadoop2 checks{color}  
-- For more information [see jdk8 (hadoop2) report|[https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340//JDK8\\_Nightly\\_Build\\_Report\\_\(Hadoop2\)/](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340//JDK8_Nightly_Build_Report_(Hadoop2)/)]

(x) {color:red}-1 jdk8 hadoop3 checks{color}  
-- For more information [see jdk8 (hadoop3) report|[https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340//JDK8\\_Nightly\\_Build\\_Report\\_\(Hadoop3\)/](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2340//JDK8_Nightly_Build_Report_(Hadoop3)/)]

(/) {color:green}+1 source release artifact{color}  
-- See build output for details.

(/) {color:green}+1 client integration test{color}

8. **hudson:** Results for branch master

[build #1523 on builds.a.o|<https://builds.apache.org/job/HBase%20Nightly/job/master/1523/>]:

(/) \*{color:green}+1 overall{color}\*

----

details (if available):

(/) {color:green}+1 general checks{color}  
-- For more information [see general report|[https://builds.apache.org/job/HBase%20Nightly/job/master/1523//General\\_Nightly\\_Build\\_Report/](https://builds.apache.org/job/HBase%20Nightly/job/master/1523//General_Nightly_Build_Report/)]

(/) {color:green}+1 jdk8 hadoop2 checks{color}  
-- For more information [see jdk8 (hadoop2) report|[https://builds.apache.org/job/HBase%20Nightly/job/master/1523//JDK8\\_Nightly\\_Build\\_Report\\_\(Hadoop2\)/](https://builds.apache.org/job/HBase%20Nightly/job/master/1523//JDK8_Nightly_Build_Report_(Hadoop2)/)]

(/) {color:green}+1 jdk8 hadoop3 checks{color}  
-- For more information [see jdk8 (hadoop3) report|[https://builds.apache.org/job/HBase%20Nightly/job/master/1523//JDK8\\_Nightly\\_Build\\_Report\\_\(Hadoop3\)/](https://builds.apache.org/job/HBase%20Nightly/job/master/1523//JDK8_Nightly_Build_Report_(Hadoop3)/)]

(/) {color:green}+1 source release artifact{color}  
-- See build output for details.

(/) {color:green}+1 client integration test{color}

9. **stack:** Removed 2.3.0 and master as target. ProcedureStore implementation has changed in these branches.

10. **anoop.hbase:** bq.And maybe it is safe to just increase the file id without deleting the broken proc wal file(this is a typical solution in WAL based system), but anyway, usually deleting a wal file is not a good idea..

[~pankaj2461] Following this suggestion also ur issue can be solved right? Want to give a patch?

11. **pankajkumar:** My bad I forgot this, will raise a PR soon.