Cloudera Runtime 7.1.9

HDFS Troubleshooting

Date published: 2023-08-31
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Handling Rollback issues with ZDU

This troubleshooting section addresses issues encountered during the rollback of a failed upgrade while performing Zero Downtime Upgrade (ZDU) involving file append and open file operations.

There are three cases identified when performing a rollback. Each case involves specific steps and circumstances that result in unexpected behavior related to block generation stamps (genstamps) in the NameNode. These cases provide root cause of these problems and solutions.

Step 1: Recover files appended during ZDU

This section provides a scenario and the root cause of the problem.

Scenario

1. Create a file and close it.
2. Perform the rolling upgrade. Do not finalize the upgrade.
3. Append to the same file created before the upgrade.
4. Roll back the upgrade to the older version.

Problem

```
     # hdfs fsck /
     /dummy/appendfile: CORRUPT blockpool BP-1519352817-172.27.13.76-1692314
082449 block blk_1073747034
     /dummy/appendfile: CORRUPT 1 blocks of total size 3302 B.
     Status: CORRUPT
     Number of data-nodes: 5
     Number of racks:  1
     Total dirs:   6
     Total symlinks:  0

     Replicated Blocks:
     Total size: 6604 B (Total open files size: 490 B)
     Total files: 2 (Files currently being written: 2)
     Total blocks (validated): 2 (avg. block size 3302 B) (Total open file
 blocks (not validated): 2)
     ********************************
     UNDER MIN REPL'D BLOCKS: 1 (50.0 %)
     MINIMAL BLOCK REPLICATION: 1
     CORRUPT FILES: 1
     CORRUPT BLOCKS:  1
     CORRUPT SIZE:  3302 B
     ********************************
     Minimally replicated blocks: 1 (50.0 %)
     Over-replicated blocks: 0 (0.0 %)
     Under-replicated blocks: 0 (0.0 %)
     Mis-replicated blocks:  0 (0.0 %)
     Default replication factor: 3
     Average block replication: 1.5
     Missing blocks:  0
     Corrupt blocks:  1
     Missing replicas:  0 (0.0 %)
     Blocks queued for replication: 0
     Erasure Coded Block Groups:
     Total size: 0 B
```

```
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
Under-erasure-coded block groups: 0
Unsatisfactory placement block groups: 0
Average block group size: 0.0
Missing block groups:  0
Corrupt block groups:  0
Missing internal blocks: 0
Blocks queued for replication: 0
FSCK ended at Wed Aug 23 00:50:49 UTC 2023 in 23 milliseconds
The filesystem under path '/' is CORRUPT
```

Append operations performed during the rolling upgrade creates block files with new generation stamps (genstamps). After the rollback, the NameNode is rolled back to its previous state and is unaware of newer genstamps created during the rolling upgrade.

The rolled-back NameNode will fail to recognize these new genstamps. This makes the NameNode to treat the upcoming genstamps as corrupt blocks. This keeps the NameNode in safe mode when you restart the NameNode during the rollback.

```
The reported blocks 4563 has reached the threshold 0.9990 of total blocks
 4566. The number of live datanodes 6 has reached the minimum number 1.
 Name node detected blocks with generation stamps in future. This means that
 Name node metadata is inconsistent. This can happen if Name node metadata
 files have been manually replaced. Exiting safe mode will cause loss of
 807114959 byte(s). Please restart name node with right metadata or use
 "hdfs dfsadmin -safemode forceExit" if you are certain that the NameNode
 was started with the correct FsImage and edit logs. If you encountered this
 during a rollback, it is safe to exit with -safemode forceExit.
```

Solution

Solution for the use case.

Recovery Steps

You can get the list of corrupt blocks using hdfs fsck /. After identifying the corrupt blocks, recover the corrupt blocks:

1. The Namenode will be in safemode at this point, We can force exit out of safemode to perform the following operations `hdfs dfsadmin -safemode forceExit`.
2. At this point the Namenode is out of safemode. You must copy the corrupt file data to a backup file `hdfs dfs -cp <corrupt_file> <backup_file>`.
3. Truncate the corrupt file to 0 size `hdfs dfs -truncate 0 <corrupt_file>`.
4. Append the backup file to the original corrupt file `hdfs dfs -cat <backup_file> | hdfs dfs -appendToFile - <corrupt_file>`.

Step 2: Recover previous files Hsync'ed during ZDU

This section provides a scenario and the root cause of the problem.

Scenario

1. Open a file and write some content.
2. Perform Hsync operation so that the metadata is persisted on the NameNode.
3. Perform a rolling upgrade. It restarts the DNs and breaks the existing pipeline.

4. Continue with writing and close the file.
5. Rollback the upgrade to the older version.

Problem

After the rollback, the NameNode is aware up to the Hsync state. However, Data Nodes have already incremented genstamps as the write continues to happen through the upgrade.

This does not lead to corrupted blocks but leads to the issue when the rolled-back NameNode detects upocoming genstamps for a file that is committed. However, the NameNode is not aware of it. This keeps the NameNode in safe mode.

You can read the content that is written before, during, and after the upgrade. This breaks the contract of reading more content than before the upgrade.

Solution

Solution for the use case.

Recovery Steps

You can read the file that is written after the upgrade.

If the file is not closed and the client dies during the upgrade. The file's complete metadata is persisted when the lease expires for the file.

You can manually recover the lease. However, you need to handle the CORRUPT files using Step 1. After handling the recovery steps, you can continue with manual lease recovery.

1. Identify the files open to write ## `hdfs fsck / -openforwrite /dummy/appendfile: CORRUPT blockpool BP-1519352817-172.27.13.76-1692314082449 block blk_1073747034 /dummy/appendfile: CORRUPT 1 blocks of total size 3302 B./dummy/text1 490 bytes, replicated: replication=2, 1 block(s), OPENFORWRITE: /dummy/text2 0 bytes, replicated: replication=2, 1 block(s), OPENFORWRITE:.`
2. At this point, NameNode is in safemode. After handling the CORRUPT files using Step 1, you can force exit out of safe mode `hdfs dfsadmin -safemode forceExit`.
3. Now, the NameNode is out of safemode. You can start recovering lease for the open files `hdfs debug recoverLease -path <open_file_path>`.

Step 3: Recover open files in corrupt state

This section provides a scenario and the root cause of the problem.

Steps to Reproduce

1. Open a file and start writing some content.
2. Perform a rolling upgrade. It restarts the DNs and hence breaks the existing pipeline.
3. Continue with writing and close the file.
4. Rollback the upgrade to the older version.

Problem

The problem is same as Step 2. The difference is you should not persist metadata until the file is complete.

Solution

Solution for the use case.

Recovery Steps

After the client stops writing or the lease expires, the data is persisted to the NameNode. You can perform a manual lease recovery using Step 2. However, you can still read the file.

Summary

Understand the summary of all the use cases, root causes, and solutions in this section.

Open file cases (Step 2 and Step 3) is recovered eventually when the lease expires and the metadata is persisted to the NameNode. However, the problem is, it can have additional data that is written after the upgrade.

The append API is the main cause of files being corrupt as you continue to write to the blocks with future genstamps. Rolling back at this point corrupts the file because of genstamp mismatch. However, using Step 1, you can recover the corrupted files.

Related Information

Rollback HDFS