



Hadoop HDFS / HDFS-5558

LeaseManager monitor thread can crash if the last block is complete but another block is not.

Details

Type:	Bug	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	0.23.9, 2.3.0	Fix Version/s:	0.23.10, 2.3.0
Component/s:	None		
Labels:	None		
Target Version/s:	0.23.11		
Hadoop Flags:	Reviewed		

Description

As mentioned in [HDFS-5557](#), if a file has its last and penultimate block not completed and the file is being closed, the last block may be completed but the penultimate one might not. If this condition lasts long and the file is abandoned, LeaseManager will try to recover the lease and the block. But `internalReleaseLease()` will fail with invalid cast exception with this kind of file.

Attachments

HDFS-5558.branch-023.patch	2 kB	02/Dec/13 20:14
HDFS-5558.branch-023.patch	2 kB	23/Nov/13 04:30
HDFS-5558.patch	1 kB	02/Dec/13 20:14
HDFS-5558.patch	0.8 kB	23/Nov/13 02:18

Issue Links

relates to

HDFS-7342 Lease Recovery doesn't happen some times	RESOLVED
--	-----------------

Activity

- [Kihwal Lee](#) added a comment - 23/Nov/13 02:11
- LeaseManager might need to be made more robust, but more importantly, the last block shouldn't be completed if the penultimate block is not completed when closing a file. In `completeFileInternal()`, `checkFileProgress(pendingFile, false)` needs to be called before calling `commitOrCompleteLastBlock()`. If the penultimate block isn't going to be completed soon, the close will fail anyway. It should fail before doing more damage.
- [Jason Darrell Lowe](#) added a comment - 23/Nov/13 03:09
- We might need a slight addition to the patch for branch-0.23, as `getAdditionalBlock` has a very similar structure where it can commit/complete the last block before checking the state of the penultimate block. If we get stuck in that state for a while or client abandons the file then the LeaseManager could hit the same condition if a block report completes the last block but not the penultimate block. In trunk/branch-2 `getAdditionalBlock` calls `analyzeFileState` before it tries to commit/complete the block, and that will throw if the penultimate block has not completed.
- [Kihwal Lee](#) added a comment - 23/Nov/13 04:30
- Posting the branch-0.23 version of the patch. PreCommit will fail on this.
- [Hadoop QA](#) added a comment - 23/Nov/13 04:33


-1 overall. Here are the results of testing the latest attachment

<http://issues.apache.org/jira/secure/attachment/12615446/HDFS-5558.branch-023.patch>
against trunk revision .

-1 patch. The patch command could not apply the patch.

Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5555//console>

This message is automatically generated.

▼  Hadoop QA added a comment - 23/Nov/13 04:36

-1 overall. Here are the results of testing the latest attachment

<http://issues.apache.org/jira/secure/attachment/12615435/HDFS-5558.patch>
against trunk revision .

+1 @author. The patch does not contain any @author tags.

-1 tests included. The patch doesn't appear to include any new or modified tests.

Please justify why no new tests are needed for this patch.

Also please list what manual steps were performed to verify this patch.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. The javadoc tool did not generate any warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.


+1 core tests. The patch passed unit tests in hadoop-hdfs-project/hadoop-hdfs.

+1 contrib tests. The patch passed contrib unit tests.

Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5552//testReport/>

Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5552//console>

This message is automatically generated.

▼  Vinayakumar B added a comment - 23/Nov/13 16:13

I tried to reproduce the problem as mentioned with the help of test changes in [HDFS-5557](#), but could not get the Invalid Cast Exception in trunk code.

Instead, **checkLeases () got stuck in infinite loop with fsn writelock held**. Because checkLeases() will check repeatedly for the files until all files are renewed. Instead get all the expired leases and check once and return the call, check again after NAMENODE_LEASE_RECHECK_INTERVAL. if required this would be a separate Jira though.

As of now I am seeing only possible case could be [HDFS-5557](#), which leads to this case.

▼  Binglin Chang added a comment - 25/Nov/13 08:20

Is this related to [HDFS-4882](#) ? It has been left there for a long time.

▼  Vinayakumar B added a comment - 25/Nov/13 09:02

[HDFS-4882](#) is more related to `dfs.namenode.replication.min=2` which is not taking care of adding extra datanode during PIPELINE_CLOSE_RECOVERY,


I agree, not adding extra datanodes can cause checkLeases () into infinite loop, but with the fix given in this jira that situation may not come.

Now client's close() have timeout and fails after timeout.


▼  Kihwal Lee added a comment - 25/Nov/13 16:03

I tried to reproduce the problem as mentioned with the help of test changes in [HDFS-5557](#), but could not get the Invalid Cast Exception in trunk code.

The test won't reproduce this issue because the client will lose every time. In order to reproduce this, the client has to lose the race in [HDFS-5557](#) for the penultimate block and the datanode has to win the race for the last block. I.e. produce block layout [...] [BlockInfoUnderConstruction:COMMITTED][BlockInfo:COMPLETE].

▼  Colin McCabe added a comment - 25/Nov/13 18:42


My understanding is that we can only get into this situation if there is another bug (such as [HDFS-5557](#)) causing an internal inconsistency. With this in mind, I think the log message should be at ERROR level, not INFO, and should look different from the standard `checkFileProgress` log message. It looks good aside from that.

▼  Kihwal Lee added a comment - 02/Dec/13 16:16


My understanding is that we can only get into this situation if there is another bug (such as [HDFS-5557](#)) causing an internal inconsistency.

Faulty or busy data nodes might delay the incremental block report for the penultimate block of a file or crash before sending it. We have also seen in the past a name node getting overwhelmed with RPC calls and falling behind in processing incremental block reports. I think it was due to a user using a small block size and creating way too many blocks (before min block size fix).


The block and replica state updates are asynchronous, so we cannot say it won't happen. In fact, we even have the close retry logic for this reason. Since it should be rare, how about making it WARN?

▼  Colin McCabe added a comment - 02/Dec/13 17:33


Thanks for the explanation. Using WARN here is fine with me.

▼  Kihwal Lee added a comment - 02/Dec/13 20:14

The patch has been updated according to the review comment.

▼  Daryn Sharp added a comment - 02/Dec/13 20:44

+1 It sounds like you addressed Colin's concern. And yes, I helped debug a case awhile back where the user accidentally set the block size to N-KB instead of N-MB for their jobs. The NN was overwhelmed and the dfs client's request for another block was being processed before the prior block's updates.

▼  Hadoop QA added a comment - 02/Dec/13 22:38

-1 overall. Here are the results of testing the latest attachment
<http://issues.apache.org/jira/secure/attachment/12616609/HDFS-5558.patch>
against trunk revision .

+1 @author. The patch does not contain any @author tags.

-1 tests included. The patch doesn't appear to include any new or modified tests.
Please justify why no new tests are needed for this patch.
Also please list what manual steps were performed to verify this patch.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. The javadoc tool did not generate any warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

-1 core tests. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs:

org.apache.hadoop.hdfs.server.balancer.TestBalancerWithNodeGroup

+1 contrib tests. The patch passed contrib unit tests.

Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5612//testReport/>

Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5612//console>


This message is automatically generated.

▼  Hudson added a comment - 03/Dec/13 11:33

FAILURE: Integrated in Hadoop-Hdfs-0.23-Build #809 (See <https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/809/>)
[HDFS-5558](#). LeaseManager monitor thread can crash if the last block is complete but another block is not. Contributed by Kihwal

Lee. (kihwal: <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1547197>)

- /hadoop/common/branches/branch-0.23/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- /hadoop/common/branches/branch-0.23/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java

▼  Kihwal Lee added a comment - 03/Dec/13 14:17

Thanks for the reviews. I've committed this to trunk, branch-2 and branch-0.23.

▼  Hudson added a comment - 03/Dec/13 14:26

SUCCESS: Integrated in Hadoop-trunk-Commit #4819 (See <https://builds.apache.org/job/Hadoop-trunk-Commit/4819/>) [HDFS-5558](#). LeaseManager monitor thread can crash if the last block is complete but another block is not. Contributed by Kihwal Lee. (kihwal: <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1547393>)

- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java

▼  Hudson added a comment - 04/Dec/13 11:00

FAILURE: Integrated in Hadoop-Yarn-trunk #411 (See <https://builds.apache.org/job/Hadoop-Yarn-trunk/411/>) [HDFS-5558](#). LeaseManager monitor thread can crash if the last block is complete but another block is not. Contributed by Kihwal Lee. (kihwal: <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1547393>)

- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java

▼  Hudson added a comment - 04/Dec/13 13:27

FAILURE: Integrated in Hadoop-Mapreduce-trunk #1628 (See <https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1628/>) [HDFS-5558](#). LeaseManager monitor thread can crash if the last block is complete but another block is not. Contributed by Kihwal Lee. (kihwal: <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1547393>)

- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java

▼  Hudson added a comment - 04/Dec/13 13:39

SUCCESS: Integrated in Hadoop-Hdfs-trunk #1602 (See <https://builds.apache.org/job/Hadoop-Hdfs-trunk/1602/>) [HDFS-5558](#). LeaseManager monitor thread can crash if the last block is complete but another block is not. Contributed by Kihwal Lee. (kihwal: <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1547393>)

- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java

▼ People

Assignee:



Kihwal Lee

Reporter:



Kihwal Lee

Votes:

0

Vote for this issue

Watchers:

11

Start watching this issue

▼ Dates

Created:

4/13/23, 3:05 PM

[HDFS-5558] LeaseManager monitor thread can crash if the last block is complete but another block is not. - ASF JIRA

23/Nov/13 02:04

Updated:

19/Nov/14 23:16

Resolved:

03/Dec/13 14:17