Hadoop HDFS / HDFS-3701

HDFS may miss the final block when reading a file opened for writing if one of the datanode is dead

Details

Type:	Bug	Status:	CLOSED
Priority:	Critical	Resolution:	Fixed
Affects Version/s:	1.0.3	Fix Version/s:	1.1.0
Component/s:	hdfs-client		
Labels:	None		
Hadoop Flags:	Reviewed		

Description

When the file is opened for writing, the DFSClient calls one of the datanode owning the last block to get its size. If this datanode is dead, the socket exception is shallowed and the size of this last block is equals to zero. This seems to be fixed on trunk, but I didn't find a related Jira. On 1.0.3, it's not fixed. It's on the same area as HDFS-1950 or ~~HDFS-3222~~.

Attachments

HDFS-3701.branch-1.v2.merged.patch	11 kB	07/Sep/12 16:48
HDFS-3701.branch-1.v3.patch	11 kB	24/Sep/12 16:02
HDFS-3701.branch-1.v4.patch	11 kB	25/Sep/12 00:45
HDFS-3701.ontopof.v1.patch	2 kB	27/Aug/12 15:40
HDFS-3701.patch	11 kB	26/Aug/12 18:39

Issue Links

duplicates

HDFS-3965 DFSInputStream should not eat up exceptions if file is under construction	RESOLVED

is related to

HDFS-3222 DFSInputStream#openInfo should not silently get the length as 0 when locations length is zero...	CLOSED
HBASE-6751 Too many retries, leading a a delay to read the HLog after a datanode failure	CLOSED

relates to

HBASE-6401 HBase may lose edits after a crash if used with HDFS 1.0.3 or older	CLOSED
HDFS-4590 Add more Unit Test Case for HDFS-3701 HDFS Miss Final Block Reading when File is Open for ...	OPEN

Activity

Uma Maheswara Rao G added a comment - 23/Jul/12 11:42

Hi Nicolas,

Thanks a lot for digging into DFS changes.

By seeing the description, ~~HDFS-3222~~ can fix partly, and there are other changes to read the block from all DNs and check specific to ReplicaNotFoundException. If it sees ReplicaNotFoundException from all DNs, then only we will return 0, otherwise we will throw exception.
Combining these changes should fix this problems. But I don't think we have the specific exception like ReplicaNotFoundException in

branch-1.
I think this new category of exceptions got introduced as part new Appned design. So, distinquishing the replica not found exception and other remote exceptions will be difficult in branch-1 (let me check the latest code, whether it is possible due to many issues backporting recently)

Thanks
Uma

Nicolas Liochon added a comment - 23/Jul/12 14:52

Hi Uma,

Thank you very much for the feedback. In 1.0.3 ReplicaNotFoundException is not there for sure. If it's not there on the branch-1, and if the backport is not planned, a possible workaround would be to at least try on all the datanodes available, and if they all fail with a socket exception rethrow an exception instead of continuing. For other cases, we would return 0.
That would not solve all cases, but would decrease the probability of occurence. And this should not bring false positive, the only case that would work today but not after this fix is the case with an empty last block and all datanodes dead. In this case, we would throw an error while previously the DFSClient would have returned O.

Michael Stack added a comment - 23/Jul/12 16:38

To be clear, this issue is about data loss.

Nicolas Liochon added a comment - 23/Jul/12 18:47

btw, I if you're ok with the approach I can propose a patch on branch-1

Uma Maheswara Rao G added a comment - 24/Jul/12 04:56

Yes, that should help us in almost solving this problem.
In our internal branch(based on branch-1), we were re-throwing the exception after trying for all the nodes.

```
To be clear, this issue is about data loss.
```

Yes, Stack. This I have seen in my clusters. We solved it by adding above proposed code and ~~HDFS-3222~~.

(That time I concentrated to fix ~~HDFS-3222~~ only on branch-2. But I should have proposed the changes for branch-1 as well ☹ . See the effect versions marked in ~~HDFS-3222~~). One small gap I have seen in branch-1 is, bytes acked not tracked properly compared to hadoop-2 today. so, if we read the length from some other node which is having lesser length than primary node, and primary node connect back just before starting the actual read request. That time, still this kind of problems will be there. I have seen other JIRA, that 'we have to mark that failed node into dead node list when we get the rpc errors while fetching the length' should help in solving that issue.
Have not seen so far after that fix in our internal branch.

So, I am +1 for doing that.

@Nicolas, do you have patch ready for branch-1? if no, I will generate the patch on branch-1 in some time next week.

Nicolas Liochon added a comment - 24/Jul/12 07:33

@Uma
I don't have the patch, so if you generate it it's obviously fine by me. Thanks a lot!

Nicolas Liochon added a comment - 09/Aug/12 14:09

Hi Uma,

Have you been able to generate the patch?

Thanks you,

Nicolas

Uma Maheswara Rao G added a comment - 09/Aug/12 14:18

Hi Nicolas, I really did not get the time to work on it as I was busy in some tasks from last few days. Hopefully I will get to it on next week or so.
BTW, If you feel urgent and you have time to do it, I would be happy to review it and commint in some time.

Nicolas Liochon added a comment - 09/Aug/12 15:21

Thanks for the quick answer, Uma. It's not a matter of days for us, so we can wait for you. And short term I will propose something for ~~HDFS-3705~~ as the workaround I have for it in HBase is a little bit too much of a workaround 🙂.

Uma Maheswara Rao G added a comment - 26/Aug/12 18:44

Attached initial version of patch, Which is basically merge of ~~HDFS-3222~~ and added the code to try from other DNs when getting the length for last block. I have not tested this, as I am putting my major efforts on Hadoop-2, did not installed branch-1 cluster.
Do you mind have a look and test the same.

Nicolas Liochon added a comment - 26/Aug/12 20:55

Thanks Uma. Sure, I will try it early next week.

Nicolas Liochon added a comment - 27/Aug/12 15:39

Hi Uma,

I've done a few changes, it seems to work. HBase tests are ok with this new HDFS version, HDFS tests are in progress locally but seems to work ok as well. ~~HDFS-3701~~.ontopof.v1.patch contains my changes only.

Tsz-wo Sze added a comment - 06/Sep/12 23:27

Hi Nicolas, which branch is your patch for? I tried to apply it for branch-1 but it failed.

Uma Maheswara Rao G added a comment - 07/Sep/12 04:18

Thanks N, for the update on patch. sorry for the late on this.

@Nicholas, I think both patches may required to apply. First we have to apply other patch and then ~~HDFS-3701~~.ontopof.v1.patch. Because he mentioned 'ontopof'.

Today, If I get time, I will merge both and provide a single one after reviewing his changes. After that it should be possible for you to apply and check. Thanks a lot, Nicholas for your time.

Nicolas Liochon added a comment - 07/Sep/12 16:45

Hi Uma and Nicholas,

I uploaded a merged and rebased patch, for branch-1. It contains nothing else that what is already in ~~HDFS-3701~~.ontopof.v1.patch and ~~HDFS-3701~~.patch.

It's in ~~HDFS-3701~~.branch-1.v2.merged.patch

Thanks for your time!

Nicolas Liochon added a comment - 21/Sep/12 09:50

Hi there,

Did you have time to look at the patch? It could cause a dataloss, so it would be great to have it integrated...

Uma Maheswara Rao G added a comment - 21/Sep/12 09:57

Hi Nicholas,

Do you mind taking a look at this patch.
Since I put the patch here, I expect some one to review this!

Seems like Nicolas, has verified it with his clusters and also updated with minor modifications in it.

Patch has: porting os ~~HDFS-3222~~ and looping over the remaining DNs to get length as trunk and hadoop-2 does.

Thanks,
Uma

Tsz-wo Sze added a comment - 21/Sep/12 14:22

Hi Uma and Nicolas, sorry that I have not got a chance to check the patch. Will review it soon.

Tsz-wo Sze added a comment - 24/Sep/12 11:37

- fetchLocatedBlocksAndGetLastBlockLength() returns the last block length but the length is only used for checking whether it equals to -1. So how about changing the return type to boolean (false means location unavailable) and renaming it to fetchLocatedBlocks()? Then, openInfo could be simplified as below

```
    synchronized void openInfo() throws IOException {
      for(int retries = 3; retries > 0; retries--) {
        if (fetchLocatedBlocks()) {
          //fetch block success
          return;
        } else {
          // Last block location unavailable. When a cluster restarts,
          // DNs may not report immediately. At this time partial block
          // locations will not be available with NN for getting the length.
          // Lets retry a few times to get the length.
          DFSClient.LOG.warn("Last block locations unavailable. "
              + "Datanodes might not have reported blocks completely."
              + " Will retry for " + retries + " times");
          waitFor(4000);
        }
      }
      throw new IOException("Could not obtain the last block locations.");
    }
```

- We may also change the return type of updateBlockInfo(..) to boolean since the length is not used. I am fine if you want to keep the length.
- Throw InterruptedIOException, a subclass of IOException, instead of IOException in waitFor(..).

Uma Maheswara Rao G added a comment - 24/Sep/12 12:00

Thanks a lot, Nicholas for the review.
I will address them in next patch soon.

Suresh Srinivas added a comment - 24/Sep/12 12:27

Uma, if you are busy let me know. I can post address Nicholas's comments. Given that 1.1.0 new RC might be built today, it is good to get this in soon.

Uma Maheswara Rao G added a comment - 24/Sep/12 16:02

Thanks a lot, Suresh for the help you offered.
I have attached a patch which addresses Nicholas's comments.

Tsz-wo Sze added a comment - 24/Sep/12 23:47

- fetchLocatedBlocks() should use the return value of updateBlockInfo(..).

```
@@ -1965,54 +1996,77 @@
       updateBlockInfo(newInfo);
       this.locatedBlocks = newInfo;
       this.currentNode = null;
+      return true;
     }
```

  The code above should be

```
@@ -1965,54 +1996,77 @@
-      updateBlockInfo(newInfo);
+      boolean b = updateBlockInfo(newInfo);
       this.locatedBlocks = newInfo;
       this.currentNode = null;
+      return b;
     }
```

- This is a separated issue: The following comment should be removed since the old and new block lists must be the same even with append. Append may add new blocks but the existing blocks must be the same. See if you also want to remove it with the patch here.

```
//In fetchLocatedBlocks(),
       // I think this check is not correct. A file could have been appended to
```

```
        // between two calls to openInfo().
```

Uma Maheswara Rao G added a comment - 25/Sep/12 00:45

Oops, Its my mistake. Prepared it on hurry. Really sorry for my mistake here.

Actually I might have removed thinking that updateBlkInfo will throw exception in failure. did not looked bak once changed updateBlkInfo.

Removed comment along with this patch.

In trunk, waitFor is throwing IOException, do you think, I can file a small trivial bug and change for consistency in code?

Tsz-wo Sze added a comment - 25/Sep/12 01:02

+1 patch looks good. Please run tests and test-patch if you haven't. Thanks a lot!

Uma Maheswara Rao G added a comment - 25/Sep/12 01:50

I have just committed to branch-1, Committed revision 1389678.
Have not seen any new failure in my env with this path.

Thanks a lot Nicolas for your contribution in this issue and closer track on it.

Thanks a lot to Nicholas for your reviews!

Matthew Foley added a comment - 28/Sep/12 21:03

merged to branch-1.1

Matthew Foley added a comment - 17/Oct/12 18:27

Closed upon release of Hadoop-1.1.0.

People

Assignee:

Nicolas Liochon

Reporter:

Nicolas Liochon

Votes:

1 Vote for this issue

Watchers:

17 Start watching this issue

Dates

Created:

23/Jul/12 10:50

Updated:

12/Mar/13 00:35

Resolved:

25/Sep/12 01:50