Hadoop Common / HADOOP-3007

DataNode pipelining : failure on mirror results in failure on upstream datanode

Details

Type:	Bug	Status:	CLOSED
Priority:	Blocker	Resolution:	Fixed
Affects Version/s:	0.16.0	Fix Version/s:	0.16.2
Component/s:	None		
Labels:	None		

Description

When datanode is transfering data to other datanodes (as opposed to a client write()), DN currently receiving the data is supposed to continued to receive even if the downstream (mirror) datanode fails.

I think there are two minor bugs in receiveBlock() :

1. mirrorSock masks this.mirrorSock, which could be set to null by other members like receiveChunk() to indicate failure.
2. when this.mirrorSock is non null, failure to write to it at in receiveBlock() should be handled same way as in receiveChunk().

I am not sure if this should block 0.16.1, but should surely block 0.16.2.

Attachments

HADOOP-3007.patch	5 kB	19/Mar/08 19:30
HADOOP-3007.patch	5 kB	18/Mar/08 21:32

Activity

Raghu Angadi created issue - 12/Mar/08 21:24

Raghu Angadi made changes - 12/Mar/08 21:26

Field	Original Value	New Value
Description	When datanode is transfering data to other datanodes (as opposed to a client write()), DN currently receiving the data is supposed to continued to receive even if the downstream (mirror) datanode fails. I think there are two minor bugs in receiveBlock() : # mirrorSock masks this.mirrorSock, which could be set to null by other members like receiveChunk() to indicate failure. # when this.mirrorSock is non null, failure to write to it at in receiveBlock() should be handled same way as in receiveChunk(). I am not sure if this should block 0.16.1, but should surely block 0.16.2.	When datanode is transfering data to other datanodes (as opposed to a client write()), DN currently receiving the data is supposed to continued to receive even if the downstream (mirror) datanode fails. I think there are two minor bugs in receiveBlock() : # mirrorSock masks this.mirrorSock, which could be set to null by other members like receiveChunk() to indicate failure. # when this.mirrorSock is non null, failure to write to it at in receiveBlock() should be handled same way as in receiveChunk(). I am not sure if this should block 0.16.1, but should surely block 0.16.2.
Summary	DataNode pipelining : failure on mirror make results in failure on upstream datanode	DataNode pipelining : failure on mirror results in failure on upstream datanode

Raghu Angadi made changes - 12/Mar/08 21:49

Attachment HADOOP-3006.patch [12377732]

Raghu Angadi made changes - 12/Mar/08 21:50

Attachment ~~HADOOP-3006~~.patch [12377732]

Raghu Angadi made changes - 12/Mar/08 21:50

Comment [Patch updated with improved comments.]

Raghu Angadi made changes - 18/Mar/08 18:46

Assignee Raghu Angadi [rangadi]

Raghu Angadi added a comment - 18/Mar/08 21:32

Attached patch does the following :

- Renames conflicting arguments for receiveBlock().
- Replaces mirrorOut's accesses with :

```
try {
  access mirrorOut;
} catch (IOException e) {
  handleMirrorOutError(e);
}
```

- Note that with ~~HADOOP-1702~~, number of places where mirrorOut is written to goes down and reduces repeat of the above pattern.
- It does not change any other behaviour. If this goes after ~~HADOOP-3033~~, the patch will be smaller since 'currentWrite' is removed there.

Raghu Angadi made changes - 18/Mar/08 21:32

Attachment HADOOP-3007.patch [12378178]

Raghu Angadi added a comment - 19/Mar/08 19:30

Updated patch to latest 0.16 branch.

Raghu Angadi made changes - 19/Mar/08 19:30

Attachment HADOOP-3007.patch [12378264]

Dhruba Borthakur added a comment - 19/Mar/08 22:43

All code looks good except the first block.

This portion of code is used when the socket connection is being set up. At this point, it is better for the connection request to fail if any one mirror fails so that the client can get a fresh new set of 3 datanodes for the pipeline and start all over again.

Raghu Angadi added a comment - 19/Mar/08 22:48

> client can get a fresh new set of 3 datanodes for the pipeline and start all over again.
This patch does not change anything when a client is writing data. This affect only when one datanode is transferring (re-replicating) block to more datanodes.

Raghu Angadi added a comment - 19/Mar/08 22:49

Conceptually, there is not much difference between failure while connecting or while writing to mirror. In both cases, it just means the transfer did not happen to mirror.

Dhruba Borthakur added a comment - 19/Mar/08 23:39

+1. Code looks good.

Raghu Angadi added a comment - 20/Mar/08 16:31

Thanks Dhruba.

Raghu Angadi made changes - 20/Mar/08 16:31

Status	Open [1]	Patch Available [10002]

Hadoop QA added a comment - 21/Mar/08 04:32

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12378264/HADOOP-3007.patch
against trunk revision 619744.

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

tests included -1. The patch doesn't appear to include any new or modified tests.
Please justify why no tests are needed for this patch.

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

javac +1. The applied patch does not generate any new javac compiler warnings.

release audit +1. The applied patch does not generate any new release audit warnings.

findbugs +1. The patch does not introduce any new Findbugs warnings.

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

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

Test results: http://hudson.zones.apache.org/hudson/job/Hadoop-Patch/2017/testReport/
Findbugs warnings: http://hudson.zones.apache.org/hudson/job/Hadoop-Patch/2017/artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html
Checkstyle results: http://hudson.zones.apache.org/hudson/job/Hadoop-Patch/2017/artifact/trunk/build/test/checkstyle-errors.html
Console output: http://hudson.zones.apache.org/hudson/job/Hadoop-Patch/2017/console

This message is automatically generated.

Raghu Angadi added a comment - 21/Mar/08 18:57

I just committed this.

Raghu Angadi made changes - 21/Mar/08 18:57

Resolution		Fixed [1]
Status	Patch Available [10002]	Resolved [5]

Hudson added a comment - 23/Mar/08 05:36

Integrated in Hadoop-trunk #436 (See http://hudson.zones.apache.org/hudson/job/Hadoop-trunk/436/)

Nigel Daley made changes - 03/Apr/08 15:42

Status	Resolved [5]	Closed [6]

Owen O'Malley made changes - 08/Jul/09 16:42

Component/s	dfs [12310710]	

Transition	Time In Source Status	Execution Times
Raghu Angadi made transition - 20/Mar/08 16:31 OPEN → PATCH AVAILABLE	7d 19h 7m	1
Raghu Angadi made transition - 21/Mar/08 18:57 PATCH AVAILABLE → RESOLVED	1d 2h 25m	1
Nigel Daley made transition - 03/Apr/08 15:42 RESOLVED → CLOSED	12d 20h 45m	1

People

Assignee:

Raghu Angadi

Reporter:

Raghu Angadi

Votes:

0 Vote for this issue

Watchers:

1 Start watching this issue

Dates

Created:

12/Mar/08 21:24

Updated:

08/Jul/09 16:42

Resolved:

21/Mar/08 18:57