Item 133

git_comments:

1. the threshold is reached or was reached before
2. SafeModeMonitor thread if smmthread is already running, the block threshold must have been reached before, there is no need to enter the safe mode again

git_commits:

1. **summary:** HDFS-5140. Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error. Contributed by Jing Zhao.
   message: HDFS-5140. Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error. Contributed by Jing Zhao. git-svn-id: https://svn.apache.org/repos/asf/hadoop/common/trunk@1518899 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error
   description: While running namenode load generator with 100 threads for 10 mins namenode was being failed over ever 2 mins. The standby namenode shut itself down as it ran out of memory and was not able to create another thread. When we searched for 'Safe mode extension entered' in the standby log it was present 55000+ times

jira_issues_comments:

1. Here is the stack trace from the standby namenode {code} 2013-08-28 08:58:45,519 INFO hdfs.StateChange (FSNamesystem.java:reportStatus(4677)) - STATE* Safe mode extension entered. The reported blocks 833 has reached the threshold 1.0000 of total blocks 833. The number of live datanodes 3 has reached the minimum number 0. Safe mode will be turned off automatically in 29 seconds. 2013-08-28 08:58:45,524 ERROR namenode.FSEditLogLoader (FSEditLogLoader.java:loadEditRecords(203)) - Encountered exception on operation CloseOp [length=0, inodeId=0, path=/user/hrt_qa/ha-loadgenerator/100-threads/dir3/dir2/dir5/dir4/dir2/dir1/hostname63, replication=3, mtime=1377680236411, atime=1377680236320, blockSize=134217728, blocks=[blk_1073940431_205511], permissions=hrt_qa:hrt_qa:rw-r--r--, clientName=, clientMachine=, opCode=OP_CLOSE, txid=1141116] java.lang.OutOfMemoryError: unable to create new native thread at java.lang.Thread.start0(Native Method) at java.lang.Thread.start(Thread.java:640) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem$SafeModeInfo.checkMode(FSNamesystem.java:4521) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem$SafeModeInfo.incrementSafeBlockCount(FSNamesystem.java:4568) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem$SafeModeInfo.access$1900(FSNamesystem.java:4275) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.incrementSafeBlockCount(FSNamesystem.java:4854) at org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.completeBlock(BlockManager.java:596) at org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.completeBlock(BlockManager.java:608) at org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.forceCompleteBlock(BlockManager.java:621) at org.apache.hadoop.hdfs.server.namenode.FSEditLogLoader.updateBlocks(FSEditLogLoader.java:696) at org.apache.hadoop.hdfs.server.namenode.FSEditLogLoader.applyEditLogOp(FSEditLogLoader.java:372) at org.apache.hadoop.hdfs.server.namenode.FSEditLogLoader.loadEditRecords(FSEditLogLoader.java:198) at org.apache.hadoop.hdfs.server.namenode.FSEditLogLoader.loadFSEdits(FSEditLogLoader.java:111) at org.apache.hadoop.hdfs.server.namenode.FSImage.loadEdits(FSImage.java:733) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer.doTailEdits(EditLogTailer.java:227) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread.doWork(EditLogTailer.java:321) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread.access$200(EditLogTailer.java:279) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread$1.run(EditLogTailer.java:296) at org.apache.hadoop.security.SecurityUtil.doAsLoginUserOrFatal(SecurityUtil.java:456) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread.run(EditLogTailer.java:292) 2013-08-28 08:58:45,597 FATAL ha.EditLogTailer (EditLogTailer.java:doWork(328)) - Unknown error encountered while tailing edits. Shutting down standby NN. java.io.IOException: Failed to apply edit log operation CloseOp [length=0, inodeId=0, path=/user/hrt_qa/ha-loadgenerator/100-threads/dir3/dir2/dir5/dir4/dir2/dir1/hostname63, replication=3, mtime=1377680236411, atime=1377680236320, blockSize=134217728, blocks=[blk_1073940431_205511], permissions=hrt_qa:hrt_qa:rw-r--r--, clientName=, clientMachine=, opCode=OP_CLOSE, txid=1141116]: error unable to create new native thread at org.apache.hadoop.hdfs.server.namenode.MetaRecoveryContext.editLogLoaderPrompt(MetaRecoveryContext.java:94) at

org.apache.hadoop.hdfs.server.namenode.FSEditLogLoader.loadEditRecords(FSEditLogLoader.java:204) at org.apache.hadoop.hdfs.server.namenode.FSEditLogLoader.loadFSEdits(FSEditLogLoader.java:111) at org.apache.hadoop.hdfs.server.namenode.FSImage.loadEdits(FSImage.java:733) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer.doTailEdits(EditLogTailer.java:227) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread.doWork(EditLogTailer.java:321) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread.access$200(EditLogTailer.java:279) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread$1.run(EditLogTailer.java:296) at org.apache.hadoop.security.SecurityUtil.doAsLoginUserOrFatal(SecurityUtil.java:456) at org.apache.hadoop.hdfs.server.namenode.ha.EditLogTailer$EditLogTailerThread.run(EditLogTailer.java:292) 2013-08-28 08:58:45,636 INFO util.ExitUtil (ExitUtil.java:terminate(124)) - Exiting with status 1 {code}

2. **body:** Looks like the problem is caused by the following code (FSNamesystem#checkMode): {code} reached = now(); smmthread = new Daemon(new SafeModeMonitor()); smmthread.start(); reportStatus("STATE* Safe mode extension entered.", true); {code} In SBN, because the block threshold keeps being adjusted while tailing the editlog, we may have the following scenarios: reach the block threshold, enter the final 30 seconds of safemode --> block threshold is adjusted, and the number of safe block cannot reach the threshold --> reach the block threshold again.... Because of the above code, each time the block threshold is met, a new safemode monitor thread will be created while the old one keeps running behind. Thus a large number of safemode monitor threads can be created. This code is fine in the active NN (or the NN in non-HA setup) because we do not adjust block threshold there and once the NN goes out of the safemode it will not go in again.
   label: code-design
3. **body:** Thus the simplest fix is to check if a non-null smmthread is already running before we start a new one in the checkMode() method. In the meanwhile, because of the 30 seconds waiting time in safemode, and the block threshold can keep being adjusted, the SBN may be in safemode for a long time. One possible solution is that once the SBN finds itself can leave the safemode, it will not enter the safemode again, even if the block threshold is increased later. But in this way, the safemode in the SBN can not cover the same transaction range as the active NN. Another way is that the SBN fetch the latest transaction id in the beginning, and keeps tracking the block number for safemode only till that transaction id. But this may add some unnecessary complexity.
   label: code-design
4. **body:** I think if SBN crosses the threshold and is in the process of moving out of safemode, it does not make sense to enter safemode again. +1 for not going back to safemode as the block count keeps changing. Other alternative solutions seem needlessly complicated at no obvious benefits.
   label: code-design
5. Thanks for the comments, Suresh. Upload the initial patch.
6. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12600642/HDFS-5140.001.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. 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. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-hdfs-project/hadoop-hdfs. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/4910//testReport/ Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/4910//console This message is automatically generated.
7. smmthread should be set to null in SafemodeMonitor holding the writeLock.
8. Update the patch to address Suresh's comment.
9. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12600695/HDFS-5140.002.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. 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. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-hdfs-project/hadoop-hdfs. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/4912//testReport/ Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/4912//console This message is automatically generated.
10. **body:** +1 for the patch. The current code in SafeModeInfo#canLeave() checks needEnter() again. This is bothersome, since in case of secondary we could flip flop about leaving and entering safemode. The whole safemode seems to have become complicated in case of secondary. We should perhaps create a jira about at least not checking needEnter() again.
    label: code-design
11. Thanks for the review, Suresh! I've committed this to trunk, branch-2 and branch-2.1-beta. Also filed HDFS-5145 to revisit the needEnter check in SafeModeInfo#canLeave.
12. SUCCESS: Integrated in Hadoop-trunk-Commit #4352 (See [https://builds.apache.org/job/Hadoop-trunk-Commit/4352/]) HDFS-5140. Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error. Contributed by Jing Zhao. (jing9: http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1518899) * /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

13. SUCCESS: Integrated in Hadoop-Yarn-trunk #317 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk/317/]) HDFS-5140. Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error. Contributed by Jing Zhao. (jing9: http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1518899) * /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
14. FAILURE: Integrated in Hadoop-Hdfs-trunk #1507 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk/1507/]) HDFS-5140. Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error. Contributed by Jing Zhao. (jing9: http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1518899) * /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
15. FAILURE: Integrated in Hadoop-Mapreduce-trunk #1534 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1534/]) HDFS-5140. Too many safemode monitor threads being created in the standby namenode causing it to fail with out of memory error. Contributed by Jing Zhao. (jing9: http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1518899) * /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