Item 144
git_comments:

1. Walk through the children of this node, using a new summary array for the (sub)tree rooted at this node
2. update the passed summary array with the values for this node's subtree
3. Create directory /hdfs-2053
4. Create a file under subdirectory B
5. Create a file under subdirectory C (which has a space quota)
6. Test HDFS-2053 :
7. Factors to vary the sizes of test files created in each subdir. The actual factors are not really important but they allow us to create identifiable file sizes per subdir, which helps during debugging.
8. Create a file under subdirectory A
9. Check space consumed for /hdfs-2053
10. Create subdirectories /hdfs-2053/{A,B,C}
11. Set space quota for subdirectory C

git_commits:

1. **summary:** HDFS-2053. Bug in INodeDirectory#computeContentSummary warning. Contributed by Michael Noll
message: HDFS-2053. Bug in INodeDirectory#computeContentSummary warning. Contributed by Michael Noll git-svn-id: https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20@1140850 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Bug in INodeDirectory#computeContentSummary warning
description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the

warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

2. **summary:** Bug in INodeDirectory#computeContentSummary warning
   description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could

apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

3. **summary:** Bug in INodeDirectory#computeContentSummary warning
   description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.
4. **summary:** Bug in INodeDirectory#computeContentSummary warning
   description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The

bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

5. **summary:** Bug in INodeDirectory#computeContentSummary warning
   description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already

contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

6. **summary:** Bug in INodeDirectory#computeContentSummary warning
   description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-

0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.
label: code-design

7. **summary:** Bug in INodeDirectory#computeContentSummary warning
description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.
label: test
8. **summary:** Bug in INodeDirectory#computeContentSummary warning
description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the

parent directory but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

9. **summary:** Bug in INodeDirectory#computeContentSummary warning
   description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the

warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

10. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could

apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

11. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.
12. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The

bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

13. **summary:** Bug in INodeDirectory#computeContentSummary warning
description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already

contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

14. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-

0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

15. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.
16. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20,

branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

17. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-

1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

18. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions

(0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

19. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20, branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.
    label: code-design
20. **summary:** Bug in INodeDirectory#computeContentSummary warning
    description: *How to reproduce* {code} # create test directories $ hadoop fs -mkdir /hdfs-1377/A $ hadoop fs -mkdir /hdfs-1377/B $ hadoop fs -mkdir /hdfs-1377/C # ...add some test data (few kB or MB) to all three dirs... # set space quota for subdir C only $ hadoop dfsadmin -setSpaceQuota 1g /hdfs-1377/C # the following two commands _on the parent dir_ trigger the warning $ hadoop fs -dus /hdfs-1377 $ hadoop fs -count -q /hdfs-1377 {code} Warning message in the namenode logs: {code} 2011-06-09 09:42:39,817 WARN org.apache.hadoop.hdfs.server.namenode.NameNode: Inconsistent diskspace for directory C. Cached: 433872320 Computed: 438465355 {code} Note that the commands are run on the _parent directory_ but the warning is shown for the _subdirectory_ with space quota. *Background* The bug was introduced by the HDFS-1377 patch, which is currently committed to at least branch-0.20,

branch-0.20-security, branch-0.20-security-204, branch-0.20-security-205 and release-0.20.3-rc2. In the patch, {{src/hdfs/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java}} was updated to trigger the warning above if the cached and computed diskspace values are not the same for a directory with quota. The warning is written by {{computecontentSummary(long[] summary)}} in {{INodeDirectory}}. In the method an inode's children are recursively walked through while the {{summary}} parameter is passed and updated along the way. {code} /** {@inheritDoc} */ long[] computeContentSummary(long[] summary) { if (children != null) { for (INode child : children) { child.computeContentSummary(summary); } } {code} The condition that triggers the warning message compares the current node's cached diskspace (via {{node.diskspaceConsumed()}}) with the corresponding field in {{summary}}. {code} if (-1 != node.getDsQuota() && space != summary[3]) { NameNode.LOG.warn("Inconsistent diskspace for directory " +getLocalName()+". Cached: "+space+" Computed: "+summary[3]); {code} However {{summary}} may already include diskspace information from other inodes at this point (i.e. from different subtrees than the subtree of the node for which the warning message is shown; in our example for the tree at {{/hdfs-1377}}, {{summary}} can already contain information from {{/hdfs-1377/A}} and {{/hdfs-1377/B}} when it is passed to inode {{/hdfs-1377/C}}). Hence the cached value for {{C}} can incorrectly be different from the computed value. *How to fix* The supplied patch creates a fresh summary array for the subtree of the current node. The walk through the children passes and updates this {{subtreeSummary}} array, and the condition is checked against {{subtreeSummary}} instead of the original {{summary}}. The original {{summary}} is updated with the values of {{subtreeSummary}} before it returns. *Unit Tests* I have run "ant test" on my patched build without any errors*. However the existing unit tests did not catch this issue for the original HDFS-1377 patch, so this might not mean anything. ;-) That said I am unsure what the most appropriate way to unit test this issue would be. A straight-forward approach would be to automate the steps in the _How to reproduce section_ above and check whether the NN logs an incorrect warning message. But I'm not sure how this check could be implemented. Feel free to provide some pointers if you have some ideas. *Note about Fix Version/s* The patch _should_ apply to all branches where the HDFS-1377 patch has committed to. In my environment, the build was Hadoop 0.20.203.0 release with a (trivial) backport of HDFS-1377 (0.20.203.0 release does not ship with the HDFS-1377 fix). I could apply the patch successfully to {{branch-0.20-security}}, {{branch-0.20-security-204}} and {{release-0.20.3-rc2}}, for instance. Since I'm a bit confused regarding the upcoming 0.20.x release versions (0.20.x vs. 0.20.20x.y) I have been so bold and added 0.20.203.0 to the list of affected versions even though it is actually only affected when HDFS-1377 is applied to it... Best, Michael *Well, I get one error for {{TestRumenJobTraces}} but first this seems to be completely unrelated and second I get the same test error when running the tests on the stock 0.20.203.0 release build.

jira_issues_comments:

1. Patch version 1 for HDFS-2053. The patch should apply to all branches to which the original HDFS-1377 patch has been applied to. See the ticket description for more details regarding "Fix Version/s".
2. Again, I am not sure how to properly identify the correct names of the versions. For instance, the patch successfully applies to branch-0.20-security-204 but I am not sure whether this translates to version "0.20.204.0" in the dropdown list.
3. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12481915/HDFS-2053_v1.txt against trunk revision 1133476. +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 patch. The patch command could not apply the patch. Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/750//console This message is automatically generated.
4. New patch version, no properly using 'git diff --no-prefix' to generate it. Doh!
5. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12481918/HDFS-2053_v2.txt against trunk revision 1133476. +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 patch. The patch command could not apply the patch. Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/751//console This message is automatically generated.
6. **body:** Hey Michael - thank you for the excellent report! In summary, the condition used to warn in FSDirectory#computeContentSummary has a bug, it compares the cached value for the directory not to a

computed value for that directory but to a computed value that includes the directory and it's siblings. The bug results in a spurious warning, it doesn't impact eg the correctness of quotas. Given this I think two things are reasonable: # Remove the warning (which removes the bug) # Compute the correct summary for just that directory (your patch) The latter sounds good to me. Allocating a 4 long array for each level in the directory hierarchy isn't bad and this method isn't on a hot path. Nit, I'd change array allocation to the following since we assume summary has len 4 and should be faster. {noformat} assert 4 == summary.length; long[] subtreeSummary = new long[]{0,0,0,0} {noformat} Wrt testing how about right after space is calculated adding the following: {noformat} assert -1 == node.getDsQuota() || space == subtreeSummary[3]; {noformat} Asserts are enabled by default when the tests are run, if TestQuota doesn't trigger this assert then add a test similar to what you did manullay which will trigger it. Also, please generate a patch against trunk (HDFS-2053_v2.txt doesn't apply for me). Thanks!
label: code-design

7. **body:** Hi Eli, many thanks for your quick reply and feedback! FYI: I have integrated your suggestions into the patch. Currently I am waiting for "ant test" to finish to see whether {{TestQuota}} does indeed trigger this assert, and thus whether I actually need to add any special test to it. @unit testing: From what I have seen {{TestQuota#testSpaceCommands()}} would be the place to add a test for this issue, and {{dfs.getContentSummary().getSpaceConsumed(Path)}} would be the way to "indirectly" check the diskspace consumed by a directory and its children. It seems to be semantically equivalent to the actual {{INodeDirectory#computeContentSummary(long[])}} method I want to test but it appears to be several layers up the call stack [1]. Is this correct? If so my test case using {{dfs.getContentSummary()}} would be basically: 1. Create parent dir + 3 subdirs {{A,B,C}}. 2. {{DFSTestUtil.createFile()}} a file of {a,b,c} * {{fileLength}} in {{A,B,C}}, respectively. 3. Test whether {{getSpaceConsumed()}} of {{A,B,C}} equals the expected value, i.e. {a,b,c} * {{fileLength}} * {{replication}}. 4. Test whether {{getSpaceConsumed()}} of the parent dir equals {a+b+c} * {{fileLength}} * {{replication}}. I'm just asking since I'm not fully sure whether {{dfs.getContentSummary()}} would not mask etc. any hidden issues in the "lower level" method {{INodeDirectory.computeContentSummary()}}. Best, Michael [1] {{FSDirectory#getContentSummary(String)}} seems to be the method that actually calls {{InodeDirectory#computeContentSummary(long[])}} at some point.
   label: test
8. Apologies for the slow reply! You are correct about the test case, I think what you suggest would catch it. Will review the new patch when you post it. Thanks, Eli
9. New version of the patch based on trunk, integrating Eli's feedback and unit test.
10. +1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12483937/HDFS-2053_v3.txt against trunk revision 1140030. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +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 core unit tests. +1 contrib tests. The patch passed contrib unit tests. +1 system test framework. The patch passed system test framework compile. Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/847//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/847//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/847//console This message is automatically generated.
11. +1 looks good
12. Patch for branch-20 attached.
13. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12484473/hdfs-2053_v3-b20.patch against trunk revision 1140707. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. -1 patch. The patch command could not apply the patch. Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/861//console This message is automatically generated.
14. Integrated in Hadoop-Hdfs-trunk-Commit #763 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk-Commit/763/]) HDFS-2053. Bug in INodeDirectory#computeContentSummary warning. Contributed by Michael Noll eli : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1140707 Files : * /hadoop/common/trunk/hdfs/CHANGES.txt * /hadoop/common/trunk/hdfs/src/test/hdfs/org/apache/hadoop/hdfs/TestQuota.java * /hadoop/common/trunk/hdfs/src/java/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java

15. I've committed this to trunk and branch-20 (after running the tests). Thanks Micahel!
16. You're welcome. Thanks for the quick integration, Eli!
17. Integrated in Hadoop-Hdfs-trunk #710 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk/710/]) HDFS-2053. Bug in INodeDirectory#computeContentSummary warning. Contributed by Michael Noll eli : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1140707 Files : * /hadoop/common/trunk/hdfs/CHANGES.txt * /hadoop/common/trunk/hdfs/src/test/hdfs/org/apache/hadoop/hdfs/TestQuota.java * /hadoop/common/trunk/hdfs/src/java/org/apache/hadoop/hdfs/server/namenode/INodeDirectory.java
18. Eli, can you please commit this to 0.20.s. This will be picked up when 0.20.205 branch is created.
19. I've merged this to branch-0.20-security.
20. Closed upon release of 0.20.205.0