Item 236
git_comments:

1. * * Creates a new VolumeFailureInfo. * * @param failedStorageLocation storage location that has failed * @param failureDate date/time of failure in milliseconds since epoch * @param estimatedCapacityLost estimate of capacity lost in bytes
2. * * Tracks information about failure of a data volume.
3. * * Creates a new VolumeFailureInfo, when the capacity lost from this volume * failure is unknown. Typically, this means the volume failed immediately at * startup, so there was never a chance to query its capacity. * * @param failedStorageLocation storage location that has failed * @param failureDate date/time of failure in milliseconds since epoch
4. * * Returns the storage location that has failed. * * @return storage location that has failed
5. * * Returns estimate of capacity lost. This is said to be an estimate, because * in some cases it's impossible to know the capacity of the volume, such as if * we never had a chance to query its capacity before the failure occurred. * * @return estimate of capacity lost in bytes
6. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
7. * * Returns date/time of failure * * @return date/time of failure in milliseconds since epoch
8. * * Returns the date/time of the last volume failure in milliseconds since * epoch. * * @return date/time of last volume failure in milliseconds since epoch
9. * * Summarizes information about data volume failures on a DataNode.
10. * * Returns each storage location that has failed, sorted. * * @return each storage location that has failed, sorted
11. * * Returns estimate of capacity lost. This is said to be an estimate, because * in some cases it's impossible to know the capacity of the volume, such as if * we never had a chance to query its capacity before the failure occurred. * * @return estimate of capacity lost in bytes
12. * * Creates a new VolumeFailureSummary. * * @param failedStorageLocations storage locations that have failed * @param lastVolumeFailureDate date/time of last volume failure in * milliseconds since epoch * @param estimatedCapacityLostTotal estimate of capacity lost in bytes
13. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
14. * * Returns info about volume failures. * * @return info about volume failures, possibly null
15. optimization. Recent versions of the DataNode report a VolumeFailureSummary containing the date/time of the last volume failure. If that's available, then we check that instead for greater accuracy.
16. * * Returns info about volume failures. * * @return info about volume failures, possibly null
17. FsDatasetSpi
18. FSDatasetMBean
19. * * Gets initial volume failure information for all volumes that failed * immediately at startup. The method works by determining the set difference * between all configured storage locations and the actual storage locations in * use after attempting to put all of them into service. * * @return each storage location that has failed
20. Tracks volume failures, sorted by volume path.
21. * * Returns each storage location that has failed, sorted. * @return each storage location that has failed, sorted
22. * * Returns an estimate of total capacity lost due to volume failures in bytes. * @return estimate of total capacity lost in bytes
23. * * Returns the date/time of the last volume failure in milliseconds since * epoch. * @return date/time of last volume failure in milliseconds since epoch
24. FSNamesystemMBean

25. * * Returns an estimate of total capacity lost due to volume failures in bytes * across all live data nodes. * @return estimate of total capacity lost in bytes
26. * * Number of failed data volumes across all live data nodes. * @return number of failed data volumes across all live data nodes
27. Filter the display to only datanodes with volume failures.
28. FsDatasetSpi
29. FSDatasetMBean
30. Create file1 and wait for 3 replicas (ie all DNs can still store a block). Then assert that all DNs are up, despite the volume failures.
31. * * Returns expected capacity lost for use in assertions. The return value is * dependent on whether or not it is expected that the volume capacities were * known prior to the failures. * * @param expectCapacityKnown if true, then expect that the capacities of the * volumes were known before the failures, and therefore the lost capacity * can be reported * @param expectedVolumeFailuresTotal expected number of failed volumes * @return estimated capacity lost in bytes
32. Reconfigure again to try to add back the failed volumes.
33. * * Checks a DataNode for correct reporting of failed volumes. * * @param dn DataNode to check * @param expectedVolumeFailuresCounter metric counter value for * VolumeFailures. The current implementation actually counts the number * of failed disk checker cycles, which may be different from the length of * expectedFailedVolumes if multiple disks fail in the same disk checker * cycle * @param expectCapacityKnown if true, then expect that the capacities of the * volumes were known before the failures, and therefore the lost capacity * can be reported * @param expectedFailedVolumes expected locations of failed volumes * @throws Exception if there is any failure
34. NN sees reduced capacity, but no volume failures.
35. These tests simulate volume failures by denying execute permission on the volume's path. On Windows, the owner of an object is always allowed access, so we can't run these tests on Windows.
36. Make the first two volume directories on the first two datanodes non-accessible.
37. * * Initializes the cluster. * * @param numDataNodes number of datanodes * @param storagesPerDatanode number of storage locations on each datanode * @param failedVolumesTolerated number of acceptable volume failures * @throws Exception if there is any failure
38. * * Reconfigure a DataNode by setting a new list of volumes. * * @param dn DataNode to reconfigure * @param newVols new volumes to configure * @throws Exception if there is any failure
39. Fail the first volume on both datanodes (we have to keep the third healthy so one node in the pipeline will not fail).
40. This can be thrown if reconfiguration tries to use a failed volume. We need to swallow the exception, because some of our tests want to cover this case.
41. Reconfigure each DataNode to remove its failed volumes.
42. Ensure we wait a sufficient amount of time
43. * * Checks NameNode tracking of a particular DataNode for correct reporting of * failed volumes. * * @param dm DatanodeManager to check * @param dn DataNode to check * @param expectCapacityKnown if true, then expect that the capacities of the * volumes were known before the failures, and therefore the lost capacity * can be reported * @param expectedFailedVolumes expected locations of failed volumes * @throws Exception if there is any failure
44. Ensure we wait a sufficient amount of time.
45. Calculate the total capacity of all the datanodes. Sleep for three seconds to be sure the datanodes have had a chance to heartbeat their capacities.
46. Restore executable permission on all directories where a failure may have been simulated by denying execute access. This is based on the maximum number of datanodes and the maximum number of storages per data node used throughout the tests in this suite.
47. The NN reports two volume failures
48. The NN reports two volume failures again.
49. Bring up two more datanodes
50. Eventually the NN should report four volume failures
51. Reinitialize the cluster, configured with 4 storage locations per DataNode and tolerating up to 2 failures.
52. * * Checks the NameNode for correct values of aggregate counters tracking failed * volumes across all DataNodes. * * @param expectCapacityKnown if true, then expect that the capacities of the * volumes were known before the failures, and therefore the lost capacity * can be reported * @param expectedVolumeFailuresTotal expected number of failed volumes
53. Reconfigure a third time with the failed volumes. Afterwards, we expect the same volume failures to be reported. (No double-counting.)

git_commits:

1. **summary:** HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth.
   message: HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth.

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Track and display failed DataNode storage locations in NameNode.
   description: During heartbeats, the DataNode can report a list of its storage locations that have been taken out of service due to failure (such as due to a bad disk or a permissions problem). The NameNode can track these failed storage locations and then report them in JMX and the NameNode web UI.

jira_issues_comments:

1. The attached screenshot shows the proposed appearance in the NameNode web UI. The "Failed Volumes" counter column has been replaced with "Failed Storage Locations", which shows a list of each actual location that has failed. We'll still need to retain the old counter for backwards compatibility in the JMX metrics, but I chose to remove it from the web UI display. I'm also attaching a first cut prototype patch. I'm not going to click Submit Patch yet, because I still need to write new tests for this.
2. Failed counter column is quite useful for analysis as one can sort that column. Perhaps we should add another column with details of failed storages. Is there a number that summarizes failed volume count across the cluster?
3. [~sureshms], thank you for the feedback. That's a good observation about sorting. The table is not sortable directly in the web UI right now, so I assume you mean something like copy-pasting into a text file or spreadsheet where you can sort externally. I've attached screenshot #2 showing both the old failed volume counter and the new locations column. I'll explore giving the table built-in sort functionality too on the side. That might be an easy thing with a jQuery plugin. On the overview tab, we can add Total Failed Volumes to the Summary section. I've attached screenshot #3 to show that. It's also a hyperlink that jumps to the Datanode Information tab, just like the other Summary fields related to datanodes.
4. I found a bug in {{MiniDFSCluster}} that breaks one of the tests I'm adding as part of this patch. The fix is a mid-sized change of its own, and mostly unrelated to the changes here, so I filed it as a separate issue: HDFS-7632. I've linked that as a pre-requisite, and the patch is already attached on HDFS-7632 for review.
5. I'd like to defer the work of sorting by column in the web UI to a separate jira, in order to simplify code review here for the failed volume reporting. I filed issue HDFS-7636 for the sorting.
6. Actually, HDFS-6407 is an existing jira that already tracks sorting in the UI.
7. The attached patch implements the feature. Summary: * The protocol definition for heartbeat requests has been changed to add {{failedStorageLocations}}, which contains multiple strings used to report the local file system path of each failed storage. * The DN calculates its failed storage locations as the set difference between everything configured in {{dfs.datanode.data.dir}} and the current live volumes in use by the {{FsDatasetImpl}}. Doing it this way works well with the live DN reconfiguration feature (HDFS-6808), because it will use the current active configuration rather than what was loaded at process start time. * The failed storage locations are exposed through {{FSDatasetMBean}}, so the metrics on an individual DN will publish that information. I also updated {{FsDatasetImpl#getNumFailedVolumes}} to keep its implementation in sync with the new method. * {{FsVolumeList}} no longer needs to track a separate counter of failed volumes. As a side effect, I believe this is fixing a potential bug with live DN reconfiguration. (If a previously failed volume was brought back online through live reconfiguration, then I don't believe this counter would have been decremented or reset to reflect the new state.) * On the NN side, the heartbeat handling now updates its data structures to keep track of the failed storage locations per DN. * The failed storage locations for all DNs are exposed through {{FSNamesystemMBean}}. There is also a new counter for the total volume failures across all DNs. * The web UI templates have been updated to display the new data. * {{TestDataNodeVolumeFailureReporting}} contains the testing related to this feature. I took the opportunity

to do a few other minor cleanups in this file. * Numerous other test files contain minor changes to deal with method signature changes related to passing the new field in the heartbeat.

8. I received some offline feedback from [~sureshms] that this might make the Datanode Information table too cluttered. I agree. I'll work on a version that moves the information into a new view, separate from the existing table.
9. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12692820/HDFS-7604.001.patch against trunk revision ec4389c. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 16 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1235 javac compiler warnings (more than the trunk's current 1208 warnings). {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.TestLeaseRecovery2 Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/9252//testReport/ Javac warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/9252//artifact/patchprocess/diffJavacWarnings.txt Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/9252//console This message is automatically generated.
10. I've done another mock-up of the UI. This version avoids adding clutter to the existing Datanodes page and instead moves failure information to its own dedicated page. Just like in the existing screenshot 3, there is a new field on the summary for Total Failed Volumes. I also intend to display lost capacity in parentheses next to it. However, unlike last time, the existing Datanodes page is unchanged. Instead, the volume failure information is on a new Datanode Volumes page, shown in new screenshot 4. This is hyperlinked from both the Total Failed Volumes field in the summary and a new tab in the top nav. The new page has a table displaying only the DataNodes that have volume failures. For each one, it displays the address, seconds since last contact, time of last volume failure, number of failed volumes, estimated capacity lost due to these volume failures, and a list of every failed storage location's path. I say that the capacity lost is an estimate, because there are going to be some edge cases that could prevent us from displaying accurate information here. For example, if a volume has an I/O error before we get a chance to check its capacity, then it's unknown how much storage is available on that volume. The end user workflow I imagine for this is that an admin first checks the summary information and notices a non-zero count for failed volumes. Then, the admin navigates to the Datanode Volumes page to get a list of volume failures across the cluster. This view lists only the DataNodes with volume failures, so the admin won't need to scan through the master list looking for individual nodes with a non-zero volume failure count. This can act as a sort of work queue for the admin recovering or replacing disks. I have not updated the patch. I need to rework the heartbeat information to provide this data for the UI. Meanwhile, Last Failure Time and Estimated Capacity Lost are displayed as TODO in the screenshot. Further feedback is welcome while I continue coding a new patch.
11. Thanks [~cnauroth] for working on this, which will be very useful for admin. The changes looks good to me with one comment about the UI: The title of the new page "Datanode Volumes" is a little bit confusing as it only shows failed volumes. A more accurate title could be "Failed Datanode Volumes". An alternative is to keep the "Datanode Volumes" as title but display all the volumes with their status in a dedicate column (e.g., RED for failed ones and Green for healthy ones). This way, admin can have a complete view of all the volumes and their status of the cluster. However, display all the volumes in a large cluster with thousands of nodes/volumes could be slow. UI should give the option to filter what to display, e.g. only failed volumes or all volumes.
12. Thank you for taking a look, Xiaoyu. I actually had the same thought as you a few days ago and updated my code to change the name of the page. I think changing the name is preferable to displaying all volumes of all nodes. The existing Datanode Information page already gives us a view of all nodes, and I'd like to keep the new page focused on just the nodes with problems, so that an admin can use it as a task list for routine cluster maintenance.
13. I'm attaching patch v002. I've also attached screenshots 5-7 showing what it looks like in the web UI now. It's still very similar to the change summary I gave in one of my prior comments. The differences are: * The UI now uses a separate page to track nodes with volume failures. The existing DataNode information page is unchanged. On the Datanode Volume Failures page, you'll see only nodes that currently have volume failures, and the details. * The internal tracking and the heartbeat protocol now uses a {{VolumeFailureSummary}} message. This tracks more detailed information than the prior version, which was just a list of strings.
14. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12697576/HDFS-7604.002.patch against trunk revision af08425. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 18 new or modified test files. {color:red}-1 javac{color}.

The applied patch generated 1163 javac compiler warnings (more than the trunk's current 1155 warnings). {color:red}-1 javadoc{color}. The javadoc tool appears to have generated 1 warning messages. See https://builds.apache.org/job/PreCommit-HDFS-Build/9505//artifact/patchprocess/diffJavadocWarnings.txt for details. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 1 new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-hdfs-project/hadoop-hdfs. Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/9505//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/9505//artifact/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html Javac warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/9505//artifact/patchprocess/diffJavacWarnings.txt Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/9505//console This message is automatically generated.

15. Patch v004 adds another test: {{TestDataNodeVolumeFailureReporting#testDataNodeReconfigureWithVolumeFailures}}. This covers use of the DataNode live reconfiguration feature with the presence of failed volumes. For example, if reconfiguration removes a previously failed data directory, then we expect to stop reporting that location as a failed volume. To get this working completely, there are also some additional changes in {{FsDatasetImpl}} and {{FsVolumeList}}. I also cleaned up javac and javadoc warnings. The Findbugs warning is unrelated.
16. The patch looks pretty good, only a couple of comments: {code} + boolean checkFailedStorages = (volFailures > volumeFailures) || {code} If volumeFailureSummary is not null, it might be more accurate to compare last failure timestamp? {code} public int getVolumeFailures() { - return volumeFailures; + return volumeFailureSummary != null ? + volumeFailureSummary.getFailedStorageLocations().length : 0; + } {code} In case of rolling upgrades, the older version of datanodes, will not send volumeFailureSummary, and the newer namenode might erroneously conclude 0 volume failures.
17. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12697854/HDFS-7604.004.patch against trunk revision 3f5431a. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 18 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 1 new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.web.TestWebHDFS Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/9519//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/9519//artifact/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/9519//console This message is automatically generated.
18. Jitendra, thank you for reviewing. Here is patch v005, containing both of the changes that you suggested. bq. If volumeFailureSummary is not null, it might be more accurate to compare last failure timestamp? Yes, that's particularly relevant when considering the new live DataNode reconfiguration feature. If volumes are reconfigured, and there are the same number of volume failures, but the actual volumes are different, then the old logic wouldn't have caught it. Comparing the last failure timestamps handles it well. bq. In case of rolling upgrades, the older version of datanodes, will not send volumeFailureSummary, and the newer namenode might erroneously conclude 0 volume failures. That's a great catch. I restored explicit tracking of the {{volumeFailures}} counter in {{DatanodeDescriptor}}. The implementation of {{DatanodeDescriptor#getVolumeFailures}} is fine for both old and new DataNode heartbeats, because for the new case, we guarantee that this counter is consistent with the value returned from {{getVolumeFailureSummary}}. The test failure in the last Jenkins run was unrelated.
19. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12697936/HDFS-7604.005.patch against trunk revision 7c6b654. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 18 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 1 new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-hdfs-project/hadoop-hdfs. Test results:

https://builds.apache.org/job/PreCommit-HDFS-Build/9528//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/9528//artifact/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/9528//console This message is automatically generated.

20. The Findbugs warning is unrelated and will be fixed by HDFS-7753.
21. +1 for the latest patch.
22. Thank you for the review, Jitendra. As a heads-up, I'm going to have to ask you to look at a separate branch-2 patch. There are some recent DataNode changes that are in trunk only, and that causes conflicts for this patch on branch-2. Part of the problem is related to HDFS-7496, and I've asked a question on that issue. I'll need resolution on that question before I can post a branch-2 patch.
23. All of the dependent patches that I was waiting on for merge to branch-2 have been committed. I'm uploading patch v006. This is just a rebase after those other patches. There are no additional code changes since Jitendra's +1 on v005. The same patch works for both trunk and branch-2.
24. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12698863/HDFS-7604.006.patch against trunk revision 2efb234. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 18 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.TestRollingUpgrade The following test timeouts occurred in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.server.datanode.TestDataNodeRollingUpgrade org.apache.hadoop.hdfs.server.namenode.TestEditLogRace Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/9582//testReport/ Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/9582//console This message is automatically generated.
25. [~cnauroth], thanks for the due diligence to ensure a clean merge to branch-2. +1 for the latest patch.
26. Jitendra, thank you again for the reviews. I have committed this to trunk and branch-2.
27. The test failures in the last run appear to be unrelated. I could not repro locally.
28. FAILURE: Integrated in Hadoop-trunk-Commit #7122 (See [https://builds.apache.org/job/Hadoop-trunk-Commit/7122/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-

hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java

29. SUCCESS: Integrated in Hadoop-Hdfs-trunk-Java8 #97 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk-Java8/97/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java *

hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java

30. FAILURE: Integrated in Hadoop-Yarn-trunk #841 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk/841/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-

hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java

31. SUCCESS: Integrated in Hadoop-Yarn-trunk-Java8 #107 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk-Java8/107/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java *

hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java

32. SUCCESS: Integrated in Hadoop-Hdfs-trunk #2039 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk/2039/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java * hadoop-hdfs-project/hadoop-

hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto

33. FAILURE: Integrated in Hadoop-Mapreduce-trunk #2058 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk/2058/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-

hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java

34. FAILURE: Integrated in Hadoop-Mapreduce-trunk-Java8 #108 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Java8/108/]) HDFS-7604. Track and display failed DataNode storage locations in NameNode. Contributed by Chris Nauroth. (cnauroth: rev 9729b244de50322c2cc889c97c2ffb2b4675cf77) * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailureReporting.java * hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNameNodePrunesMissingStorages.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/extdataset/ExternalDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeProtocol.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/metrics/FSNamesystemMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java *

hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestFsDatasetCache.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.html * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsVolumeList.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestFsDatasetImpl.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyConsiderLoad.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java * hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs/dfshealth.js * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/metrics/FSDatasetMBean.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestStorageReport.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/VolumeFailureSummary.java * hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/VolumeFailureInfo.java * hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java