

git_comments:

1. **** Main method for the tool.** * @return 0 if success, 1 for bad args. 2 if job aborted with an exception, * 3 if mr job was unsuccessful
2. active mob area
3. Use when we start this job as the base point for file "recency".
4. job failed
5. **** we are doing this ourselves instead of using FSUtils.getReferenceFilePaths because** * we know the mob region never splits, so we can only have HFileLink references * and looking for just them is cheaper then listing everything. **** This glob should match the naming convention for HFileLinks to our referenced hfile.** * As simplified explanation those file names look like "table=region-hfile". For details * see the {@link HFileLink#createHFileLinkName HFileLink implementation}.
6. check whether the current user is the same one with the owner of hbase root
7. **comment:** TODO make family and table optional
label: requirement
8. Results that mean something is incorrect
9. **** reuses the passed Text key.** appends the configured separator and then a comma separated list * of base64 encoded row keys
10. **** Scans a given table + CF for all mob reference cells to get the list of backing mob files.** * For each referenced file we attempt to verify that said file is on the FileSystem in a place * that the MOB system will look when attempting to resolve the actual value. **** The job includes counters that can help provide a rough sketch of the mob data.** **** <pre>** * Map-Reduce Framework * Map input records=10000 * ... * Reduce output records=99 * ... * CELLS PER ROW * Number of rows with 1s of cells per row=10000 * MOB * NUM_CELLS=52364 * PROBLEM * Affected rows=338 * Problem MOB files=2 * ROWS WITH PROBLEMS PER FILE * Number of HFiles with 100s of affected rows=2 * SIZES OF CELLS * Number of cells with size in the 10,000s of bytes=627 * Number of cells with size in the 100,000s of bytes=51392 * Number of cells with size in the 1,000,000s of bytes=345 * SIZES OF ROWS * Number of rows with total size in the 100,000s of bytes=6838 * Number of rows with total size in the 1,000,000s of bytes=3162 * **</pre>** **** Map-Reduce Framework:Map input records - the number of rows with mob references** **** Map-Reduce Framework:Reduce output records - the number of unique hfiles referenced** **** MOB:NUM_CELLS - the total number of mob reference cells** **** PROBLEM:Affected rows - the number of rows that reference hfiles with an issue** **** PROBLEM:Problem MOB files - the number of unique hfiles that have an issue** **** CELLS PER ROW: - this counter group gives a histogram of the order of magnitude of the * number of cells in a given row by grouping by the number of digits used in each count.** * This allows us to see more about the distribution of cells than what we can determine * with just the cell count and the row count. In this particular example we can see that * all of our rows have somewhere between 1 - 9 cells. **** ROWS WITH PROBLEMS PER FILE: - this counter group gives a histogram of the order of * magnitude of the number of rows in each of the hfiles with a problem.** e.g. in the * example there are 2 hfiles and they each have the same order of magnitude number of rows, * specifically between 100 and 999. **** SIZES OF CELLS: - this counter group gives a histogram of the order of magnitude of * the size of mob values according to our reference cells.** e.g. in the example above we * have cell sizes that are all between 10,000 bytes and 9,999,999 bytes. From this * histogram we can also see that *_most_ cells are 100,000 - 999,000 bytes and the smaller * and bigger ones are outliers making up less than 2% of mob cells. **** SIZES OF ROWS: - this counter group gives a histogram of the order of magnitude of the * size of mob values across each row according to our reference cells.** In the example above * we have rows that are are between 100,000 bytes and 9,999,999 bytes. We can also see that * about 2/3rd of our rows are 100,000 - 999,999 bytes. **** Generates a report that gives one file status per line, with tabs dividing fields.** **** <pre>** * RESULT OF LOOKUP FILE REF comma separated, base64 encoded rows when there's a problem * **</pre>** **** e.g.** **** <pre>** * MOB DIR 09c576e28a65ed2ead0004d192ffaa382019110184b30a1c7e034573bf8580aef8393402 * MISSING FILE 28e252d7f013973174750d483d358fa020191101f73536e7133f4cd3ab1065edf588d509 MmJiMjMyYzBiMTNjNzc0OTY1ZWY4NTU4ZjBmYmQ2MTU0NTIz,MmEzOGE0YTgzMTZjNDIiNW4MzM1MTdjNDVzMzEwNzAtODg= * **</pre>** **** Possible results are listed; the first three indicate things are working properly.** **** MOB DIR - the reference is in the normal MOB area for the given table and CF** **** HLINK TO ARCHIVE FOR SAME TABLE - the reference is present in the archive area for this * table and CF** **** HLINK TO ARCHIVE FOR OTHER TABLE - the reference is present in a different table and CF, * either in the MOB or archive areas (e.g. from a snapshot restore or clone)** **** ARCHIVE WITH HLINK BUT NOT FROM OUR TABLE - the reference is currently present in the archive * area for this table and CF, but it is kept there because a *_different_ table has a * reference to it (e.g. from a snapshot clone).** If these other tables are removed then * the file will likely be deleted unless there is a snapshot also referencing it. **** ARCHIVE BUT NO HLINKS - the reference is currently present in the archive for this table and * CF, but there are no references present to prevent its removal. Unless it is newer than * the general TTL (default 5 minutes) or referenced in a snapshot it will be subject to * cleaning.** **** ARCHIVE BUT FAILURE WHILE CHECKING HLINKS - Check the job logs to see why things failed while * looking for why this file is being kept around.** **** MISSING FILE - We couldn't find the reference on the FileSystem. Either there is dataloss due * to a bug in the MOB storage system or the MOB storage is damaged but in an edge case that * allows it to work for now. You can verify which by doing a raw reference scan to get the * referenced hfile and check the underlying filesystem. See the ref guide section on mob * for details.** **** HLINK BUT POINT TO MISSING FILE - There is a pointer in our mob area for this table and CF * to a file elsewhere on the FileSystem, however the file it points to no longer exists.** **** MISSING FILE BUT FAILURE WHILE CHECKING HLINKS - We could not find the referenced file, * however you should check the job logs to see why we couldn't check to see if there is a * pointer to the referenced file in our archive or another table's archive or mob area.** *
11. **** 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.**
12. **** Returns the string representation of the given number after grouping it * into log10 buckets.** e.g. 0-9 -> 1, 10-99 -> 10, ..., 100,000-999,999 -> 100,000, etc.
13. **comment:** for use in the reducer. easier than re-parsing it out of the scan string.
label: code-design
14. If a scanner caching value isn't set, pick a smaller default since we know we're doing a full table scan and don't want to impact other clients badly.
15. Results that mean data is probably already gone
16. archive area - is there an hlink back reference (from a snapshot from same table)
17. check for an hlink in the active mob area (from a snapshot of a different table)
18. Results that mean things are fine
19. **comment:** TODO confirm there aren't tags
label: requirement
20. Do not retrieve the mob data when scanning

git_commits:

1. **summary:** HBASE-23239 Reporting on status of backing MOB files from client-facing cells (#785)
message: HBASE-23239 Reporting on status of backing MOB files from client-facing cells (#785) * Adds a new MapReduce job that builds a report of health of mob files * Also builds background information on mob system use * add a basic mob architecture in the ref guide to explain

how mob takes the reference hfile value and finds the actual cell contents * add a troubleshooting mob section to the ref guide to explain how to do a mob reference scan. Signed-off-by: Peter Somogyi <psomogyi@apache.org>

github_issues:

github_issues_comments:

github_pulls:

1. **title:** HBASE-23239 Reporting on status of backing MOB files from client-facing cells
body: * Adds a new MapReduce job that builds a report of health of mob files * Also builds background information on mob system use

github_pulls_comments:

1. :broken_heart: *-1 overall** | Vote | Subsystem | Runtime | Comment | |----:|-----:|-----:|-----:| | :blue_heart: | reexec | 1m 9s | Docker mode activated. | ||| _ Prechecks _ | | :green_heart: | dupname | 0m 0s | No case conflicting files found. | | :green_heart: | hbaseanti | 0m 0s | Patch does not have any anti-patterns. | | :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. | | :yellow_heart: | test4tests | 0m 0s | 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. | ||| _ master Compile Tests _ | | :blue_heart: | mvninstall | 0m 33s | Maven dependency ordering for branch | | :green_heart: | mvninstall | 5m 46s | master passed | | :green_heart: | compile | 1m 24s | master passed | | :green_heart: | checkstyle | 1m 50s | master passed | | :green_heart: | shadedjars | 5m 2s | branch has no errors when building our shaded downstream artifacts. | | :green_heart: | javadoc | 0m 52s | master passed | | :blue_heart: | spotbugs | 4m 21s | Used deprecated FindBugs config; considering switching to SpotBugs. | | :broken_heart: | findbugs | 4m 19s | hbase-server in master has 1 extant findbugs warnings. | ||| _ Patch Compile Tests _ | | :blue_heart: | mvninstall | 0m 14s | Maven dependency ordering for patch | | :green_heart: | mvninstall | 5m 28s | the patch passed | | :green_heart: | compile | 1m 23s | the patch passed | | :green_heart: | javac | 1m 23s | the patch passed | | :broken_heart: | checkstyle | 0m 19s | hbase-mapreduce: The patch generated 12 new + 1 unchanged - 0 fixed = 13 total (was 1) | | :broken_heart: | whitespace | 0m 0s | The patch 2 line(s) with tabs. | | :green_heart: | shadedjars | 5m 4s | patch has no errors when building our shaded downstream artifacts. | | :green_heart: | hadoopcheck | 17m 32s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. | | :green_heart: | javadoc | 0m 51s | the patch passed | | :broken_heart: | findbugs | 0m 57s | hbase-mapreduce generated 1 new + 0 unchanged - 0 fixed = 1 total (was 0) | ||| _ Other Tests _ | | :broken_heart: | unit | 242m 19s | hbase-server in the patch failed. | | :green_heart: | unit | 21m 36s | hbase-mapreduce in the patch passed. | | :green_heart: | asflicense | 0m 54s | The patch does not generate ASF License warnings. | ||| 331m 27s ||| Reason | Tests | |-----:|-----:| FindBugs | module:hbase-mapreduce | | Exception is caught when Exception is not thrown in org.apache.hadoop.hbase.mob.mapreduce.MobRefReporter.run(String[]) At MobRefReporter.java:is not thrown in org.apache.hadoop.hbase.mob.mapreduce.MobRefReporter.run(String[]) At MobRefReporter.java:[line 465] | | Subsystem | Report/Notes | |-----:|-----:| | Docker | Client=19.03.4 Server=19.03.4 base: https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/artifact/out/Dockerfile | | GITHUB PR | https://github.com/apache/hbase/pull/785 | | Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile | | uname | Linux 9385301c1281 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64 GNU/Linux | | Build tool | maven | | Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR_PR-785/out/precommit/personality/provided.sh | | git revision | master / c58e80f6e6 | | Default Java | 1.8.0_181 | | findbugs | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/artifact/out/branch-findbugs-hbase-server-warnings.html | | checkstyle | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/artifact/out/diff-checkstyle-hbase-mapreduce.txt | | whitespace | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/artifact/out/whitespace-tabs.txt | | findbugs | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/artifact/out/new-findbugs-hbase-mapreduce.html | | unit | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/artifact/out/patch-unit-hbase-server.txt | | Test Results | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/testReport/ | | Max. process+thread count | 5466 (vs. ulimit of 10000) | | modules | C: hbase-server hbase-mapreduce U: . | | Console output | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/1/console | | versions | git=2.11.0 maven=2018-06-17T18:33:14Z | findbugs=3.1.11 | | Powered by | Apache Yetus 0.11.0 | https://yetus.apache.org | This message was automatically generated.
2. **body:** > whitespace | 0m 0s | The patch 2 line(s) with tabs. `` hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:127: * RESULT OF LOOKUP FILE REF comma seperated, base64 encoded rows when there's a problem hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:133: * MOB DIR 09c576e28a65ed2ead0004d192ffaa382019110184b30a1c7e034573bf8580aef8393402 `` these are examples of report output that contains tabs, so I'd rather leave them for accuracy.
label: code-design
3. * rebased to current master. * addressed qabot feedback (with the exception of not changing the example output) * addressed most of @cbaenziger's feedback.
4. :broken_heart: *-1 overall** | Vote | Subsystem | Runtime | Comment | |----:|-----:|-----:|-----:| | :blue_heart: | reexec | 1m 24s | Docker mode activated. | ||| _ Prechecks _ | | :green_heart: | dupname | 0m 0s | No case conflicting files found. | | :green_heart: | hbaseanti | 0m 0s | Patch does not have any anti-patterns. | | :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. | | :yellow_heart: | test4tests | 0m 0s | 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. | ||| _ master Compile Tests _ | | :blue_heart: | mvninstall | 0m 35s | Maven dependency ordering for branch | | :green_heart: | mvninstall | 6m 7s | master passed | | :green_heart: | compile | 1m 33s | master passed | | :green_heart: | checkstyle | 1m 55s | master passed | | :green_heart: | shadedjars | 5m 30s | branch has no errors when building our shaded downstream artifacts. | | :green_heart: | javadoc | 1m 8s | master passed | | :blue_heart: | spotbugs | 5m 23s | Used deprecated FindBugs config; considering switching to SpotBugs. | | :green_heart: | findbugs | 6m 21s | master passed | ||| _ Patch Compile Tests _ | | :blue_heart: | mvninstall | 0m 15s | Maven dependency ordering for patch | | :green_heart: | mvninstall | 5m 58s | the patch passed | | :green_heart: | compile | 1m 33s | the patch passed | | :green_heart: | javac | 1m 33s | the patch passed | | :broken_heart: | checkstyle | 0m 19s | hbase-mapreduce: The patch generated 2 new + 1 unchanged - 0 fixed = 3 total (was 1) | | :broken_heart: | whitespace | 0m 0s | The patch 2 line(s) with tabs. | | :green_heart: | shadedjars | 5m 18s | patch has no errors when building our shaded downstream artifacts. | | :green_heart: | hadoopcheck | 17m 45s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. | | :green_heart: | javadoc | 1m 1s | the patch passed | | :green_heart: | findbugs | 6m 33s | the patch passed | ||| _ Other Tests _ | | :broken_heart: | unit | 250m 12s | hbase-server in the patch failed. | | :green_heart: | unit | 21m 16s | hbase-mapreduce in the patch passed. | | :green_heart: | asflicense | 1m 0s | The patch does not generate ASF License warnings. | ||| 344m 48s ||| Reason | Tests | |-----:|-----:| Failed junit tests | hadoop.hbase.master.assignment.TestMergeTableRegionsProcedure | | Subsystem | Report/Notes | |-----:|-----:| | Docker | Client=19.03.4 Server=19.03.4 base: https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/2/artifact/out/Dockerfile | | GITHUB PR | https://github.com/apache/hbase/pull/785 | | Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile | | uname | Linux 16faff6dea32 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64 GNU/Linux | | Build tool | maven | | Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR_PR-785/out/precommit/personality/provided.sh | | git revision | master / f0f7fae400 | | Default Java | 1.8.0_181 | | checkstyle | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/2/artifact/out/diff-checkstyle-hbase-mapreduce.txt | | whitespace | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/2/artifact/out/whitespace-tabs.txt | | unit | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/2/artifact/out/patch-unit-hbase-server.txt | | Test Results |

- <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/2/testReport/> | Max. process+thread count | 5289 (vs. ulimit of 10000) | modules | C: hbase-server hbase-mapreduce U: . | Console output | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/2/console> | versions | git=2.11.0 maven=2018-06-17T18:33:14Z) findbugs=3.1.11 | Powered by | Apache Yetus 0.11.0 <https://yetus.apache.org> | This message was automatically generated.
5. **body:** Checkstyle complaints > .hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:125: * RESULT OF LOOKUP FILE REF comma separated, base64 encoded rows when there's a problem:20: File contains tab characters (this is the first instance). [FileTabCharacter] > .hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:132: * MISSING FILE 28e252d7f013973174750d483d358fa020191101f73536e7133f4cd3ab1065edf588d509 MmJiMjMyYzBiMTNjNzc0OTY1ZWY4NTU4ZjBmYmQ2MTUuNTlzMmE0OGE0YTgzMTZjNDIiNWE4MzM1MTdjNDVzMzEwNzAtODg=: Line is longer than 100 characters (found 196). [LineLength] Whitespace complaints > hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:125: * RESULT OF LOOKUP FILE REF comma separated, base64 encoded rows when there's a problem > hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:131: * MOB DIR 09c576e28a65ed2ead0004d192ffaa382019110184b30a1c7e034573bf8580aef8393402 These are all literal examples of the report output. I'd like to keep them accurate.
label: code-design
6. > Failed junit tests | hadoop.hbase.master.assignment.TestMergeTableRegionsProcedure as far as I can tell this unit test failure is unrelated.
7. that push was just rebasing to current master
8. :broken_heart: *-1 overall** | Vote | Subsystem | Runtime | Comment | |----|-----|-----|:-----| +0 :ok: | reexec | 1m 9s | Docker mode activated. | ||| _ Prechecks _ | +1 :green_heart: | dupname | 0m 0s | No case conflicting files found. | +1 :green_heart: | hbaseanti | 0m 0s | Patch does not have any anti-patterns. | +1 :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. | -0 :warning: | test4tests | 0m 0s | 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. | ||| _ master Compile Tests _ | +0 :ok: | mvnndep | 0m 34s | Maven dependency ordering for branch | +1 :green_heart: | mvninstall | 5m 48s | master passed | +1 :green_heart: | compile | 1m 25s | master passed | +1 :green_heart: | checkstyle | 1m 46s | master passed | +1 :green_heart: | shadedjars | 5m 6s | branch has no errors when building our shaded downstream artifacts. | +1 :green_heart: | javadoc | 0m 52s | master passed | +0 :ok: | spotbugs | 4m 31s | Used deprecated FindBugs config: considering switching to SpotBugs. | +1 :green_heart: | findbugs | 5m 20s | master passed | ||| _ Patch Compile Tests _ | +0 :ok: | mvnndep | 0m 14s | Maven dependency ordering for patch | +1 :green_heart: | mvninstall | 5m 32s | the patch passed | +1 :green_heart: | compile | 1m 25s | the patch passed | +1 :green_heart: | javac | 1m 25s | the patch passed | -1 :x: | checkstyle | 0m 19s | hbase-mapreduce: The patch generated 2 new + 1 unchanged - 0 fixed = 3 total (was 1) | -1 :x: | whitespace | 0m 0s | The patch 2 line(s) with tabs. | +1 :green_heart: | shadedjars | 5m 3s | patch has no errors when building our shaded downstream artifacts. | +1 :green_heart: | hadoopcheck | 17m 26s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. | +1 :green_heart: | javadoc | 0m 57s | the patch passed | +1 :green_heart: | findbugs | 6m 26s | the patch passed | ||| _ Other Tests _ | +1 :green_heart: | unit | 241m 41s | hbase-server in the patch passed. | -1 :x: | unit | 27m 25s | hbase-mapreduce in the patch failed. | +1 :green_heart: | asflicense | 1m 13s | The patch does not generate ASF License warnings. | ||| 338m 25s | Reason | Tests | |-----|:-----| Failed junit tests | hadoop.hbase.snapshot.TestExportSnapshot | Subsystem | Report/Notes | |-----|:-----| Docker | Client=19.03.5 Server=19.03.5 base: <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/3/artifact/out/Dockerfile> | GITHUB PR | <https://github.com/apache/hbase/pull/785> | Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile | uname | Linux 84f75a1b8661 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64 GNU/Linux | Build tool | maven | Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR_PR-785/out/precommit/personality/provided.sh | git revision | master / 834ccb4bf6 | Default Java | 1.8.0_181 | checkstyle | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/3/artifact/out/diff-checkstyle-hbase-mapreduce.txt> | whitespace | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/3/artifact/out/whitespace-tabs.txt> | unit | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/3/artifact/out/patch-unit-hbase-mapreduce.txt> | Test Results | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/3/testReport/> | Max. process+thread count | 5357 (vs. ulimit of 10000) | modules | C: hbase-server hbase-mapreduce U: . | Console output | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/3/console> | versions | git=2.11.0 maven=2018-06-17T18:33:14Z) findbugs=3.1.11 | Powered by | Apache Yetus 0.11.1 <https://yetus.apache.org> | This message was automatically generated.
9. again, rebase to current master.
10. :broken_heart: *-1 overall** | Vote | Subsystem | Runtime | Comment | |----|-----|-----|:-----| +0 :ok: | reexec | 1m 11s | Docker mode activated. | ||| _ Prechecks _ | +1 :green_heart: | dupname | 0m 1s | No case conflicting files found. | +1 :green_heart: | hbaseanti | 0m 0s | Patch does not have any anti-patterns. | +1 :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. | -0 :warning: | test4tests | 0m 0s | 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. | ||| _ master Compile Tests _ | +0 :ok: | mvnndep | 0m 37s | Maven dependency ordering for branch | +1 :green_heart: | mvninstall | 5m 35s | master passed | +1 :green_heart: | compile | 1m 32s | master passed | +1 :green_heart: | checkstyle | 1m 44s | master passed | +1 :green_heart: | shadedjars | 4m 53s | branch has no errors when building our shaded downstream artifacts. | +1 :green_heart: | javadoc | 0m 55s | master passed | +0 :ok: | spotbugs | 4m 16s | Used deprecated FindBugs config: considering switching to SpotBugs. | +1 :green_heart: | findbugs | 5m 4s | master passed | ||| _ Patch Compile Tests _ | +0 :ok: | mvnndep | 0m 14s | Maven dependency ordering for patch | +1 :green_heart: | mvninstall | 5m 16s | the patch passed | +1 :green_heart: | compile | 1m 26s | the patch passed | +1 :green_heart: | javac | 1m 26s | the patch passed | -1 :x: | checkstyle | 0m 21s | hbase-mapreduce: The patch generated 2 new + 1 unchanged - 0 fixed = 3 total (was 1) | -1 :x: | whitespace | 0m 0s | The patch 2 line(s) with tabs. | +1 :green_heart: | shadedjars | 4m 48s | patch has no errors when building our shaded downstream artifacts. | +1 :green_heart: | hadoopcheck | 17m 28s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. | +1 :green_heart: | javadoc | 1m 1s | the patch passed | +1 :green_heart: | findbugs | 6m 6s | the patch passed | ||| _ Other Tests _ | -1 :x: | unit | 288m 3s | hbase-server in the patch failed. | +1 :green_heart: | unit | 24m 48s | hbase-mapreduce in the patch passed. | +1 :green_heart: | asflicense | 1m 20s | The patch does not generate ASF License warnings. | ||| 381m 30s | Reason | Tests | |-----|:-----| Failed junit tests | hadoop.hbase.master.procedure.TestProcedurePriority | hadoop.hbase.replication.mutlwal.TestReplicationEndpointWithMultipleAsyncWAL | Subsystem | Report/Notes | |-----|:-----| Docker | Client=19.03.5 Server=19.03.5 base: <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/4/artifact/out/Dockerfile> | GITHUB PR | <https://github.com/apache/hbase/pull/785> | Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile | uname | Linux 6547d11a4082 4.15.0-60-generic #67-Ubuntu SMP Thu Aug 22 16:55:30 UTC 2019 x86_64 GNU/Linux | Build tool | maven | Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR_PR-785/out/precommit/personality/provided.sh | git revision | master / 9c82a65b29 | Default Java | 1.8.0_181 | checkstyle | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/4/artifact/out/diff-checkstyle-hbase-mapreduce.txt> | whitespace | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/4/artifact/out/whitespace-tabs.txt> | unit | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/4/artifact/out/patch-unit-hbase-server.txt> | Test Results | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/4/testReport/> | Max. process+thread count | 5471 (vs. ulimit of 10000) | modules | C: hbase-server hbase-mapreduce U: . | Console output | <https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/4/console> | versions | git=2.11.0 maven=2018-06-17T18:33:14Z) findbugs=3.1.11 | Powered by | Apache Yetus 0.11.1 <https://yetus.apache.org> | This message was automatically generated.
11. :broken_heart: *-1 overall** | Vote | Subsystem | Runtime | Comment | |----|-----|-----|:-----| +0 :ok: | reexec | 1m 13s | Docker mode activated. | ||| _ Prechecks _ | +1 :green_heart: | dupname | 0m 1s | No case conflicting files found. | +1 :green_heart: | hbaseanti | 0m 0s | Patch

does not have any anti-patterns. || +1 :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. || -0 :warning: | test4tests | 0m 0s | 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. || | _ master Compile Tests _ || +0 :ok: | mvnDep | 0m 13s | Maven dependency ordering for branch || +1 :green_heart: | mvninstall | 5m 28s | master passed || +1 :green_heart: | compile | 3m 16s | master passed || +1 :green_heart: | checkstyle | 2m 48s | master passed || +0 :ok: | refguide | 6m 10s | branch has no errors when building the reference guide. See footer for rendered docs, which you should manually inspect. || +1 :green_heart: | shadedjars | 5m 3s | branch has no errors when building our shaded downstream artifacts. || +1 :green_heart: | javadoc | 3m 41s | master passed || +0 :ok: | spotbugs | 15m 3s | Used deprecated FindBugs config; considering switching to SpotBugs. || +1 :green_heart: | findbugs | 20m 15s | master passed || | _ Patch Compile Tests _ || +0 :ok: | mvnDep | 0m 13s | Maven dependency ordering for patch || +1 :green_heart: | mvninstall | 5m 26s | the patch passed || +1 :green_heart: | compile | 3m 17s | the patch passed || +1 :green_heart: | javac | 3m 17s | the patch passed || -1 :x: | checkstyle | 2m 51s | root: The patch generated 5 new + 2 unchanged - 0 fixed = 7 total (was 2) || -1 :x: | whitespace | 0m 0s | The patch 2 line(s) with tabs. || +0 :ok: | refguide | 6m 17s | patch has no errors when building the reference guide. See footer for rendered docs, which you should manually inspect. || +1 :green_heart: | shadedjars | 4m 58s | patch has no errors when building our shaded downstream artifacts. || +1 :green_heart: | hadoopcheck | 21m 5s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. || +1 :green_heart: | javadoc | 4m 27s | the patch passed || +1 :green_heart: | findbugs | 22m 30s | the patch passed || | _ Other Tests _ || -1 :x: | unit | 241m 40s | root in the patch failed. || +1 :green_heart: | asflicense | 1m 41s | The patch does not generate ASF License warnings. || | | 370m 32s || Subsystem | Report/Notes | |-----|:-----| Docker | Client=19.03.5 Server=19.03.5 base: https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/artifact/out/Dockerfile || GITHUB PR | https://github.com/apache/hbase/pull/785 || Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile refguide || uname | Linux 30e8eb0a86c2 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64 GNU/Linux || Build tool | maven || Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR-785/out/precommit/personality/provided.sh || git revision | master / 60d9430195 || Default Java | 1.8.0_181 || refguide | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/artifact/out/branch-site/book.html || checkstyle | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/artifact/out/diff-checkstyle-root.txt || whitespace | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/artifact/out/whitespace-tabs.txt || refguide | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/artifact/out/patch-site/book.html || unit | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/artifact/out/patch-unit-root.txt || Test Results | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/testReport/ || Max. process+thread count | 4650 (vs. ulimit of 10000) || modules | C: hbase-server hbase-mapreduce . U: . || Console output | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/6/console || versions | git=2.11.0 maven=2018-06-17T18:33:14Z findbugs=3.1.11 || Powered by | Apache Yetus 0.11.1 https://yetus.apache.org | This message was automatically generated.

12. this last commit implements some feedback I got from someone in our support organization. essentially, they suggested I make sure each individual line about sizes was a clear stand alone statement to minimize the chances that someone would misinterpret the numbers.
13. :broken_heart: *-1 overall* | Vote | Subsystem | Runtime | Comment | |----|:-----|:-----|:-----| +0 :ok: | reexec | 1m 10s | Docker mode activated. || | _ Prechecks _ || +1 :green_heart: | dupname | 0m 0s | No case conflicting files found. || +1 :green_heart: | hbaseanti | 0m 0s | Patch does not have any anti-patterns. || +1 :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. || -0 :warning: | test4tests | 0m 0s | 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. || | _ master Compile Tests _ || +0 :ok: | mvnDep | 0m 38s | Maven dependency ordering for branch || +1 :green_heart: | mvninstall | 5m 42s | master passed || +1 :green_heart: | compile | 3m 18s | master passed || +1 :green_heart: | checkstyle | 2m 52s | master passed || +0 :ok: | refguide | 6m 27s | branch has no errors when building the reference guide. See footer for rendered docs, which you should manually inspect. || +1 :green_heart: | shadedjars | 5m 2s | branch has no errors when building our shaded downstream artifacts. || +1 :green_heart: | javadoc | 3m 39s | master passed || +0 :ok: | spotbugs | 4m 5s | Used deprecated FindBugs config; considering switching to SpotBugs. || +1 :green_heart: | findbugs | 20m 20s | master passed || | _ Patch Compile Tests _ || +0 :ok: | mvnDep | 0m 13s | Maven dependency ordering for patch || +1 :green_heart: | mvninstall | 5m 28s | the patch passed || +1 :green_heart: | compile | 3m 19s | the patch passed || +1 :green_heart: | javac | 3m 19s | the patch passed || -1 :x: | checkstyle | 2m 54s | root: The patch generated 5 new + 2 unchanged - 0 fixed = 7 total (was 2) || -1 :x: | whitespace | 0m 0s | The patch 2 line(s) with tabs. || +0 :ok: | refguide | 6m 29s | patch has no errors when building the reference guide. See footer for rendered docs, which you should manually inspect. || +1 :green_heart: | shadedjars | 5m 55s | patch has no errors when building our shaded downstream artifacts. || +1 :green_heart: | hadoopcheck | 22m 5s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. || +1 :green_heart: | javadoc | 4m 28s | the patch passed || +1 :green_heart: | findbugs | 22m 18s | the patch passed || | _ Other Tests _ || -1 :x: | unit | 260m 11s | root in the patch failed. || +1 :green_heart: | asflicense | 1m 21s | The patch does not generate ASF License warnings. || | | 393m 17s || Subsystem | Report/Notes | |-----|:-----| Docker | Client=19.03.5 Server=19.03.5 base: https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/artifact/out/Dockerfile || GITHUB PR | https://github.com/apache/hbase/pull/785 || Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile refguide || uname | Linux 165a0f8d61c8 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64 GNU/Linux || Build tool | maven || Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR-785/out/precommit/personality/provided.sh || git revision | master / 82e155eb26 || Default Java | 1.8.0_181 || refguide | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/artifact/out/branch-site/book.html || checkstyle | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/artifact/out/diff-checkstyle-root.txt || whitespace | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/artifact/out/whitespace-tabs.txt || refguide | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/artifact/out/patch-site/book.html || unit | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/artifact/out/patch-unit-root.txt || Test Results | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/testReport/ || Max. process+thread count | 4548 (vs. ulimit of 10000) || modules | C: hbase-server hbase-mapreduce . U: . || Console output | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/7/console || versions | git=2.11.0 maven=2018-06-17T18:33:14Z findbugs=3.1.11 || Powered by | Apache Yetus 0.11.1 https://yetus.apache.org | This message was automatically generated.
14. have time to take another look @cbaenziger ?
15. **body:** these three checkstyle things are new and I'll push a fix for them shortly. `` .hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:160: * * MISSING FILE BUT FAILURE WHILE CHECKING HLINKS - We could not find the referenced file, however: Line is longer than 100 characters (found 102). [LineLength] .hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:225: final Text INCONSISTENT_ARCHIVE_BAD_LINK = new Text("ARCHIVE WITH HLINK BUT NOT FROM OUR TABLE");: Line is longer than 100 characters (found 101). [LineLength] .hbase-mapreduce/src/main/java/org/apache/hadoop/hbase/mob/mapreduce/MobRefReporter.java:355: LOG.error("Exception while checking mob area of our table for HFileLinks that point to {}");: Line is longer than 100 characters (found 101). [LineLength] `` the other two are the previously mentioned example report output. > whitespace | 0m 0s | The patch 2 line(s) with tabs. these are example report output, so I think the tabs should be left in place. `` [ERROR] Failures: [ERROR] TestReplicationEndpointWithMultipleAsyncWAL>TestReplicationEndpoint.testInterClusterReplication:235 Waiting timed out after [30,000] msec Failed to replicate all edits, expected = 2500 replicated = 2467 [WARNING] Flakes: [WARNING] org.apache.hadoop.hbase.namespace.TestNamespaceAuditor.testRegionMerge(org.apache.hadoop.hbase.namespace.TestNamespaceAuditor) [ERROR] Run 1: TestNamespaceAuditor.testRegionMerge:387 Å Timeout [INFO] Run 2: PASS `` these failures in the unit tests are not related AFAICT.

label: code-design

16. :broken_heart: **:1 overall** | Vote | Subsystem | Runtime | Comment | :----:|-----:|-----:|-----:| +0 :ok: | reexec | 1m 14s | Docker mode activated. ||| _ Prechecks _ | +1 :green_heart: | dupname | 0m 0s | No case conflicting files found. | +1 :green_heart: | hbaseanti | 0m 0s | Patch does not have any anti-patterns. | +1 :green_heart: | @author | 0m 0s | The patch does not contain any @author tags. | -0 :warning: | test4tests | 0m 0s | 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. ||| _ master Compile Tests _ | +0 :ok: | mvnDep | 0m 40s | Maven dependency ordering for branch | +1 :green_heart: | mvninstall | 6m 3s | master passed | +1 :green_heart: | compile | 3m 20s | master passed | +1 :green_heart: | checkstyle | 2m 55s | master passed | +0 :ok: | refguide | 6m 30s | branch has no errors when building the reference guide. See footer for rendered docs, which you should manually inspect. | +1 :green_heart: | shadedjars | 5m 10s | branch has no errors when building our shaded downstream artifacts. | +1 :green_heart: | javadoc | 3m 48s | master passed | +0 :ok: | spotbugs | 4m 8s | Used deprecated FindBugs config; considering switching to SpotBugs. | +1 :green_heart: | findbugs | 21m 6s | master passed ||| _ Patch Compile Tests _ | +0 :ok: | mvnDep | 0m 14s | Maven dependency ordering for patch | +1 :green_heart: | mvninstall | 5m 42s | the patch passed | +1 :green_heart: | compile | 3m 18s | the patch passed | +1 :green_heart: | javac | 3m 18s | the patch passed | -1 :x: | checkstyle | 2m 57s | root: The patch generated 2 new + 2 unchanged - 0 fixed = 4 total (was 2) | -1 :x: | whitespace | 0m 1s | The patch 2 line(s) with tabs. | +0 :ok: | refguide | 6m 16s | patch has no errors when building the reference guide. See footer for rendered docs, which you should manually inspect. | +1 :green_heart: | shadedjars | 5m 5s | patch has no errors when building our shaded downstream artifacts. | +1 :green_heart: | hadoopcheck | 17m 16s | Patch does not cause any errors with Hadoop 2.8.5 2.9.2 or 3.1.2. | +1 :green_heart: | javadoc | 4m 40s | the patch passed | +1 :green_heart: | findbugs | 23m 16s | the patch passed ||| _ Other Tests _ | -1 :x: | unit | 284m 14s | root in the patch failed. | +1 :green_heart: | asflicense | 1m 34s | The patch does not generate ASF License warnings. ||| 413m 19s ||| Reason | Tests | |-----:|-----:| Failed junit tests | hadoop.hbase.io.hfile.TestCacheOnWrite ||| hadoop.hbase.client.TestFromClientSideWithCoprocessor ||| hadoop.hbase.client.TestFromClientSide ||| Subsystem | Report/Notes | |-----:|-----:| Docker | Client=19.03.5 Server=19.03.5 base: https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/artifact/out/Dockerfile ||| GITHUB PR | https://github.com/apache/hbase/pull/785 ||| Optional Tests | dupname asflicense javac javadoc unit spotbugs findbugs shadedjars hadoopcheck hbaseanti checkstyle compile refguide | | uname | Linux 1acf3e45d0a0 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64 GNU/Linux ||| Build tool | maven ||| Personality | /home/jenkins/jenkins-slave/workspace/HBase-PreCommit-GitHub-PR_PR-785/out/precommit/personality/provided.sh ||| git revision | master / d561130e82 ||| Default Java | 1.8.0_181 ||| refguide | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/artifact/out/branch-site/book.html ||| checkstyle | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/artifact/out/diff-checkstyle-root.txt ||| whitespace | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/artifact/out/whitespace-tabs.txt ||| refguide | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/artifact/out/patch-site/book.html ||| unit | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/artifact/out/patch-unit-root.txt ||| Test Results | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/testReport/ ||| Max. process+thread count | 4548 (vs. ulimit of 10000) ||| modules | C: hbase-server hbase-mapreduce . U: . ||| Console output | https://builds.apache.org/job/HBase-PreCommit-GitHub-PR/job/PR-785/8/console ||| versions | git=2.11.0 maven=2018-06-17T18:33:14Z | findbugs=3.1.11 ||| Powered by | Apache Yetus 0.11.1 https://yetus.apache.org | This message was automatically generated.
17. @joshelser you self-resolved all your concerns, does that mean you're +1?

github_pulls_reviews:

- Why base64 encoded rows rather than output like HBase shell uses? (E.g. the hex encoded ASCII output.)
- I am failing to find `ONE_DAY` used elsewhere?
- body:** Nit: Would it be more clear to have if `! MobUtils.isValidMobRefCellValue(c) { continue; }` rather than a long `if` block that is the only code path here?
label: code-design
- My ignorance here, how would cell visibility tags matter for MOB file review?
- I like the taxonomy of output severity here.
- Would it make sense to cache creating the FS object in `setup()`, rather than for every record?
- Same FS question as above.
- More FS...
- body:** Nit: `we are` for those who are not as Englishfull?
label: code-design
- Would you be able to leave an example filename this glob should be matching? (I am not sure what the `/*=-*` is going for -- but I am also a MOB newbie.
- To double check my understanding of what this message is saying, we found multiple hfilelinks in the mob area to <file>. It has <number> links and we expect only one link?
- body:** Nit: `don't` to `do not`?
label: code-design
- body:** Is this a case an administrator would need to take action on or is it possibly really bad and possibly perfectly benign?
label: code-design
- body:** Nit: `doesn't` -> `does not`?
label: code-design
- body:** Nit: `doesn't` -> `does not`?
label: code-design
- Thanks for only a WARN and not a fatal error. I do not know how MOB works with snapshot but can one leverage [HBASE-18659 - "Use HDFS ACL to give user the ability to read snapshot directly on HDFS"](<https://issues.apache.org/jira/browse/HBASE-18659>) for getting all that is needed here?
- Should this perhaps be a conf look-up so one can pass in a desired value via `HConstants.HBASE_CLIENT_SCANNER_CACHING`?
- body:** base64 can be used in lots of places and the hex output from the hbase shell can basically just be used there. Would examples for using the base64 output in the hbase shell help? I have some handy.
label: code-design
- oh good catch. getting ahead of myself there.
- I personally don't think so, but I don't feel strongly here. We should output a log message if it's not a valid ref cell since we're supposed to be using this with a scan that asks for only ref cells.
- cell tags are bits of metadata attached to cells an only available on the server side ([ref](<http://hbase.apache.org/book.html#hbase.tags>)). They get used to implement cell level visibility labels, cell level acls, cell level ttls, and (relevant for us) as a part of the MOB implementation to track what the table name was when we flushed the value into the mob storage area. If we had that cell tag, we could avoid a potential false "MISSING FILE" result. it's described in the javadoc on the report results. probably including this comment here isn't helping explain that though?
- body:** originally I wanted to include this in the result descriptions for the javadocs, but it didn't flow as well there. what do you think, prioritize getting it more visible there?
label: documentation
- body:** hum. I'm not sure. `FileSystem` already includes a cache for a given URI and Configuration; we should always use the same pairs here. I don't know how efficient that cache is, but trying to avoid it without a seeing a performance issue feels like premature optimization? I also don't

- have telemetry from anywhere big yet. let me go dig a bit.
label: code-design
24. noted.
25. noted.
26. good call.
27. good idea! It's matching HFileLinks, which aren't specific to MOB so hopefully I can link to an explanation.
28. **body:** yep! The names of HFiles that contain data are supposed to be unique. An HFileLink to a file with data should thus only be able to point to that file in exactly one target table+region. So finding multiple would mean we had two different table+region combinations that each happened to get the same UUID and also needed to be referenced here in our table. Neither of those things should happen.
label: code-design
29. definitely need to take action. it's either dataloss or dataloss waiting to happen. Let me include how to differentiate.
30. **body:** Maybe? If the ACL syncing tries to keep the needed permissions in place for all active mob files, archived files, and files referenced via hlinks, but if it does then it would work. wrt snapshots, currently this report isn't made to examine a snapshot, but in principal we should be able to reuse the same logic (and I guess add a constraint that every referenced hfile should also be in the snapshot manifest).
label: code-design
31. that's a good idea. I don't have a good sense of how much optimizing matters for the ref-only scan.
32. How's that?
33. how's the added text to the MISSING FILE description on the tool output?
34. I would likely simply use ``Base64.encode()`.`Base64.decode()`` in ``hbase shell`` if that is what you are suggesting? For me in my simple operations, I usually think of regions when coming across a base64 encoded string from `[HRegionInfo.encode()]` ([https://hbase.apache.org/1.2/apidocs/org/apache/hadoop/hbase/HRegionInfo.html#encodeRegionName\(byte\[\]\)](https://hbase.apache.org/1.2/apidocs/org/apache/hadoop/hbase/HRegionInfo.html#encodeRegionName(byte[]))). When seeing a row, I think of a hex escaped ASCII representation ala HBase shell. There may not be consistency across tools though or perhaps now there is and I am ignorantly living in HBase 1.x vs HBase 2.x and master?
35. Ah nice if there's something to counter the ``if`` that helps my eye.
36. Thanks for the tag reference! I never realized they were not available off the RS.
37. **body:** I would say that JavaDoc is quite large as is. I like a really descriptive JavaDoc but wonder if this may be a candidate to have its own documentation somewhere? Otherwise, simply having the fields clearly named as `fine/incorrect/missing_data` would allow an operator to discover them in the JavaDoc fields section without more prose?
label: documentation
38. **body:** Certainly, this is not performance critical code -- and likely not a big performance sink either way. I figured you were closer to recalling if there was a cache or not and I kept noticing the idiom as I read.
label: code-design
39. Thanks; makes good sense.
40. **body:** I appreciate all the info I need is here but it is really confusing; understandably the system should not be in this state too. Perhaps, leave this with a shorter statement such as, HBase should not have this happen, please file a ticket with the HBase project if you find yourself here? Or more usable, could you provide pointer to a description of how MOB works in the reference guide (containing how the server-side tag, HFileLink's and HFiles go together) and suggest the admin should walk through the states a MOB HFileLink should be in and determine where the missing piece is to manually fix it?
label: code-design
41. Added a comment up by ``MISSING FILE``. Should it be ``would still work`` to ``will work``; maybe suggest the user do a GET on that cell to verify (or can the tool do that verification in an "online" mode)?
42. ``exception[...]`the point to ``->`` ``Exception while checking mob area of our table for HFileLinks that point to {}``?
43. Cool; was wondering and wanted to plant the seed too. I have no use case but appreciate the idea of HBASE-18659 as a way to help tenants not need ``hbase`` user privileges on HDFS.
44. **body:** It is sadly still not consistent anywhere. The thing for me is that as a base64 encoded string I'm more likely to be able to use a single set of instructions when helping someone else (vs different when it might be hex and might not) and I'm much more likely to notice if the row key gets mangled somewhere between the report running, stuff getting copied into an email or a support ticketing system somewhere, and eventually making its way to me. (Or to a mailing list) Using ``Base64.decode`` in the `hbase shell` is exactly correct and what my examples all do.
label: code-design
45. **body:** I could redo this as a bit in the reference guide on mob. There's no "troubleshooting" section for it yet.
label: documentation
46. I could do a get. Lemme see about that. Not sure if we still have info for an hbase connection once we're in the reducer.
47. **body:** Maybe the "get help" message is better placed in that message in the job's log?
label: code-design
48. **body:** Are you suggesting new output names, new Java variable names, or new prefixes to the same? We need the level of granularity already present to diagnose what's going on with a running system. So reducing the set of outputs would hinder that. I could maybe add a ``--verbose`` or ``--simplified`` flag to give the option to pick between "tell me if something's wrong" vs "tell me what is wrong". Maybe?
label: code-design
49. ``other compaction``?
50. **body:** Okay I chatted with @esteban about this a bit. One issue is that we wouldn't be able to do a simple GET because we only have a row key and a family. Doing a scan is also complicated because a given row could have multiple MOB cells that point to different files and those files could be a mix of missing and present-but-improperly accounted for. I think that means if a given impacted row gave an error when scanning, we'd still have to scan the rest of the rows. and even after getting an error on every impacted row we wouldn't know if the errors were because of a cell that pointed at this particular file or if it was because of another cell in the family that pointed at a different file. What if instead we could have the driver build a listing of all ``hfile -> table`` so that we can derive the information we're missing by not having the server side tag? We do this once and then put the result in the distributed cache. When a reducer needs to confirm that the file is gone-gone we can check the lookup. For that to have a race condition we'd need to somehow start the job then replace the table via a snapshot we took of a different table that flushed data after we built the listing. If I grab a directory listing from an example cluster it looks like this lookup table is ~11m entries, taking up about 196MB gzipped and 1.1GB uncompressed.
label: code-design
51. **body:** Interesting. I like your reasons. I only appreciate I can see if it is likely contiguous regions with ASCII keys and similarly rationalize about issues otherwise reported. I would say this question begs the code making things more directed to one solution by design but I am happy either way.
label: code-design
52. **body:** The granularity between the various possible results is great and all the messages should be kept in. Only the description for MISSING_FILE seems like too much and I would love a more thorough explanation of MOB to refer to as an admin. And architectural description would be better placed somewhere else (e.g. the reference guide or somewhere more obviously central to MOB design). For MISSING_FILE here, simply saying MOB's broken and suggesting what components are missing/misaligned and pointing someone to the more thorough architecture reference for them to debug more deeply would be my preferred approach, if I were the admin hitting this.
label: code-design

53. 🙌 if it turns out we need the "what are the regions impacted" I could make a simple follow on to get that from the set of rows. I suspect I can do that with a loop in the shell.
54. I think I can do that. let me make a pass at updated wording.
55. oh I think I got rid of that. In an early pass, we used to make sure there wasn't a concurrent major compaction happening. it's not necessary so I removed it.
56. Maybe `IOE`->`IOException`?
57. Perhaps `IOE`->`IOException`?
58. Oh also, just updating on this, I think your taxonomy on lines [225](https://github.com/apache/hbase/pull/785/files#diff-cd64a362d79cdf08690a294d3f90feedR225), [229](https://github.com/apache/hbase/pull/785/files#diff-cd64a362d79cdf08690a294d3f90feedR229) and [233](https://github.com/apache/hbase/pull/785/files#diff-cd64a362d79cdf08690a294d3f90feedR233) is nice - only it is code comments and not visible in the Java Docs. Would you think prefixing each Text object with the severity so that the JavaDoc shows its impact? For example, `HLINK_CLONE`->`OK_HLINK_CLONE`, `ARCHIVE_ERROR_IO`->`INCORRECT_ARCHIVE_ERROR_IO`, `MISSING`->`DATALOSS_MISSING`, etc.?
59. maybe "exception" or "problem" instead of "IOE" so that we're not in java jargon at all?
60. maybe "exception" or "problem" instead of "IOE" so that we're not in java jargon at all?
61. **body:** Spelling "fist four"
label: documentation
62. Would replacing the space in the name (e.g. "MOB DIR") with an underscore help in parsing the output of this job? Or maybe you don't expect folks to ever have to do that?
63. Is it intentional to not mention how to invoke MobRefReporter here?
64. n/m, I see it over in #928
65. the space shouldn't impact how consumable it is. we use tabs to delimit the pieces one needs to pull out. I'm already using this in production just directly with linux tools fine. I'm writing a java tool to parse it next and I don't expect to have an issue there either.
66. okay to fix on commit if there isn't something else to push a commit for?
67. Ah, I missed the tabs.
68. Yes

jira_issues:

1. **summary:** Reporting on status of backing MOB files from client-facing cells
description: We should be able to do some correctness testing of the mob files referenced by current cells in a mob enabled table.

jira_issues_comments:

1. Results for branch branch-2 [build #2387 on builds.a.o](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2387/): (x) *{color:red}-1 overall{color}* ---- details (if available): (/) {color:green}+1 general checks{color} -- For more information [see general report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2387//General_Nightly_Build_Report/) (x) {color:red}-1 jdk8 hadoop2 checks{color} -- For more information [see jdk8 (hadoop2) report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2387//JDK8_Nightly_Build_Report_(Hadoop2)/) (x) {color:red}-1 jdk8 hadoop3 checks{color} -- For more information [see jdk8 (hadoop3) report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2/2387//JDK8_Nightly_Build_Report_(Hadoop3)/) (/) {color:green}+1 source release artifact{color} -- See build output for details. (/) {color:green}+1 client integration test{color}
2. Results for branch branch-2.1 [build #1743 on builds.a.o](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.1/1743/): (x) *{color:red}-1 overall{color}* ---- details (if available): (/) {color:green}+1 general checks{color} -- For more information [see general report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.1/1743//General_Nightly_Build_Report/) (x) {color:red}-1 jdk8 hadoop2 checks{color} -- For more information [see jdk8 (hadoop2) report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.1/1743//JDK8_Nightly_Build_Report_(Hadoop2)/) (/) {color:green}+1 jdk8 hadoop3 checks{color} -- For more information [see jdk8 (hadoop3) report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.1/1743//JDK8_Nightly_Build_Report_(Hadoop3)/) (/) {color:green}+1 source release artifact{color} -- See build output for details. (/) {color:green}+1 client integration test{color}
3. Results for branch master [build #1570 on builds.a.o](https://builds.apache.org/job/HBase%20Nightly/job/master/1570/): (x) *{color:red}-1 overall{color}* ---- details (if available): (/) {color:green}+1 general checks{color} -- For more information [see general report](https://builds.apache.org/job/HBase%20Nightly/job/master/1570//General_Nightly_Build_Report/) (x) {color:red}-1 jdk8 hadoop2 checks{color} -- For more information [see jdk8 (hadoop2) report](https://builds.apache.org/job/HBase%20Nightly/job/master/1570//JDK8_Nightly_Build_Report_(Hadoop2)/) (x) {color:red}-1 jdk8 hadoop3 checks{color} -- For more information [see jdk8 (hadoop3) report](https://builds.apache.org/job/HBase%20Nightly/job/master/1570//JDK8_Nightly_Build_Report_(Hadoop3)/) (/) {color:green}+1 source release artifact{color} -- See build output for details. (/) {color:green}+1 client integration test{color}
4. Results for branch branch-2.2 [build #727 on builds.a.o](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/727/): (x) *{color:red}-1 overall{color}* ---- details (if available): (/) {color:green}+1 general checks{color} -- For more information [see general report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/727//General_Nightly_Build_Report/) (x) {color:red}-1 jdk8 hadoop2 checks{color} -- For more information [see jdk8 (hadoop2) report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/727//JDK8_Nightly_Build_Report_(Hadoop2)/) (x) {color:red}-1 jdk8 hadoop3 checks{color} -- For more information [see jdk8 (hadoop3) report](https://builds.apache.org/job/HBase%20Nightly/job/branch-2.2/727//JDK8_Nightly_Build_Report_(Hadoop3)/) (/) {color:green}+1 source release artifact{color} -- See build output for details. (/) {color:green}+1 client integration test{color}