Item 24
git_comments:

1. user granted with all global permission
2. user with read-only permissions
3. Wait for the ACL table to become available
4. Verify all table/namespace permissions are erased
5. Clean the _acl_ table
6. class with faulty stop() method, controlled by flag
7. * * Performs checks for reference counting w.r.t. TableAuthManager which is used by AccessController.
8. Up the handlers; this test needs more than usual.
9. user with no permissions
10. Enable security
11. Enable EXEC permission checking
12. TODO: convert this test to cover the full matrix in https://hbase.apache.org/book/appendix_acl_matrix.html creating all Scope x Permission combinations
13. user with all permissions
14. user is table owner. will have all permissions on table
15. Test deleted the table, no problem
16. Verify enableSecurity sets up what we require
17. Set up initial grants
18. * 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.
19. setup configuration
20. user with rw permissions on column family.
21. user with create table permissions alone
22. verify that superuser can create tables
23. USER_CREATE is USER_RW plus CREATE permissions
24. all others should be denied
25. * The systemUserConnection created here is tied to the system user. In case, you are planning * to create AccessTestAction, DON'T use this systemUserConnection as the 'doAs' user * gets eclipsed by the system user.
26. create a set of test users
27. user with admin rights on the column family
28. In this particular test case, we can't use SecureBulkLoadEndpoint because its doAs will fail to move a file for a random user
29. * * Releases the resources for the given TableAuthManager if the reference count is down to 0. * @param instance TableAuthManager to be released
30. * Returns a TableAuthManager from the cache. If not cached, constructs a new one. Returned * instance should be released back by calling {@link #release(TableAuthManager)}.
31. allows subsequent nodeChildrenChanged event to preempt current processing of nodeChildrenChanged event
32. there is a newer list

git_commits:

1. **summary:** HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes()
   message: HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes()

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Use separate thread for calling ZKPermissionWatcher#refreshNodes()
   description: I came off a support case (0.98.0) where main zk thread was seen doing the following: {code} at org.apache.hadoop.hbase.security.access.ZKPermissionWatcher.refreshAuthManager(ZKPermissionWatcher.java:152) at org.apache.hadoop.hbase.security.access.ZKPermissionWatcher.refreshNodes(ZKPermissionWatcher.java:135) at org.apache.hadoop.hbase.security.access.ZKPermissionWatcher.nodeChildrenChanged(ZKPermissionWatcher.java:121) at org.apache.hadoop.hbase.zookeeper.ZooKeeperWatcher.process(ZooKeeperWatcher.java:348) at org.apache.zookeeper.ClientCnxn$EventThread.processEvent(ClientCnxn.java:519) at org.apache.zookeeper.ClientCnxn$EventThread.run(ClientCnxn.java:495) {code} There were 62000 nodes under /acl due to lack of fix from HBASE-12635, leading to slowness in table creation because zk notification for region offline was blocked by the above. The attached patch separates refreshNodes() call into its own thread. Thanks to Enis and Devaraj for offline discussion.

jira_issues_comments:

1. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12754330/14370-v1.txt against master branch at commit bada19bb54a358233db2b3e23c86d215ac2dc29b. ATTACHMENT ID: 12754330 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. {color:red}-1 javac{color}. The patch appears to cause mvn compile goal to fail with Hadoop version 2.4.0. Compilation errors resume: [ERROR] Error invoking method 'get(java.lang.Integer)' in java.util.ArrayList at META-INF/LICENSE.vm[line 1627, column 22] [ERROR] Failed to execute goal org.apache.maven.plugins:maven-remote-resources-plugin:1.5:process (default) on project hbase-assembly: Error rendering velocity resource. Error invoking method 'get(java.lang.Integer)' in java.util.ArrayList at META-INF/LICENSE.vm[line 1627, column 22]: InvocationTargetException: Index: 0, Size: 0 -> [Help 1] [ERROR] [ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch. [ERROR] Re-run Maven using the -X switch to enable full debug logging. [ERROR] [ERROR] For more information about the errors and possible solutions, please read the following articles: [ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException [ERROR] [ERROR] After correcting the problems, you can resume the

build with the command [ERROR] mvn <goals> -rf :hbase-assembly Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15437//console This message is automatically generated.

2. So this was an issue with 0.98.0 involving something fixed in 0.98.9? Why make any additional changes now?
3. Using the main zk thread for refreshing AuthManager, in my opinion, is not a scalable design. The customer cluster has well over 2000 tables. A workflow constantly creates new tables. The time for running ZKPermissionWatcher#refreshNodes() is not negligible. We should free the main zk thread for processing other zookeeper notifications.
4. **body:** I suppose this makes sense for where we don't have proc-v2, but otherwise moving permission cache updates away from the zookeeper hack to proc-v2 is the correct approach to fixing the problem you describe.
   label: code-design
5. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12754332/14370-v1.txt against master branch at commit bada19bb54a358233db2b3e23c86d215ac2dc29b. ATTACHMENT ID: 12754332 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:green}+1 core tests{color}. The patch passed unit tests in . Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15438//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15438//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15438//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15438//console This message is automatically generated.
   label: code-design
6. This change would benefit 0.98 and branch-1.x where we don't have proc-v2. Moving permission cache updates away from the zookeeper would be done in another JIRA.
7. It is standard practice to fork off a thread to do the work in the zk client since the zk notification is handled by a single thread. If a zk watcher runs for a very long time, other zk notifications are just blocked waiting for the event notification thread. We have to be careful though because the ordering of the execution for these should follow the zk notification order. In the case Ted mentioned, even with HBASE-12635 fixed, the {{ZKPermissionWatcher#refreshNodes()}} does a getData() for every 2000+ child znode which is quite costly. The other notifications (like assignment notifications) are waiting, and causing assignment to take a very long time.
8. For the patch, instead of forking a thread, can we have a SingleThreadedExecutor?
9. Currently no class is using SingleThreadedExecutor What benefit does it give us ?
10. **body:** Forking a thread is expensive, so we do not use that pattern but instead use Executors with fixed thread pools. Plus, as mentioned above, we have to make sure that the refresh requests from different even notifications should execute in the same order that the zk notification comes from. Single threaded executor should be able to do that.
    label: code-design
11. See if patch v3 is better.
12. **body:** bq. See if patch v3 is better. Thanks Ted. The executor thread is not shut down, and will cause a thread leak. I was following the AtomicReference nodes, but could not get the full semantics. Did you introduce that to pass the list of nodes to the thread? Can we simplify by just passing the list of znodes directly to the thread? There may still be a race condition between nodeChildrenChanged (which now happens in the thread) and nodeDeleted, which still executes in the zk event thread, no?
    label: code-design
13. **body:** w.r.t. thread leak, have you seen the following code ? {code} + public void close() { + executor.shutdown(); {code} w.r.t. AtomicReference, the goal is for refresher thread to be interruptible. w.r.t. race condition between nodeChildrenChanged and nodeDeleted, if a table (namespace) is deleted, client would get TableNotFoundException (NamespaceNotFoundException) for future access - before ACL is checked. Do you think tighter coordination is needed between the zk thread and the refresher thread ?
    label: code-design
14. bq. w.r.t. thread leak, have you seen the following code ? Ok, missed that. bq. Do you think tighter coordination is needed between the zk thread and the refresher thread ? In theory, there maybe a race where one thread is refreshing the table auths, while the other is deleting that permission since now, they will be executing in different threads. Maybe we can make every operation (nodeCreated,nodeDeleted,nodeDataChanged,nodeChildrenChanged) to execute from the executor.
15. bq. one thread is refreshing the table auths, while the other is deleting that permission If table auth is put back by the refresher, it would be overwritten next time the table with same name is created. Before the table is created again, TableNotFoundException serves as the guard. What do you think ?
16. Looking at patch v1, the thread is created once when ZKPermissionWatcher is created - not for every refresh call.
17. bq. Looking at patch v1, the thread is created once when ZKPermissionWatcher is created - not for every refresh call. Yes, it seems that the thread never exits. This is kind of like a busy wait, no? The thread looks for whether it should refresh every 2ms without a wait / signal.
18. Clarification, based in patch v1, the only concern is busy waiting. If that's the case, I can continue refining patch v1.
19. Please take a look at 14370-wait-nofity.txt
20. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12754799/14370-wait-nofity.txt against master branch at commit e95358a7fc3f554dcbb351c8b7295cafc01e8c23. ATTACHMENT ID: 12754799 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:green}+1 core tests{color}. The patch passed unit tests in . Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15485//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15485//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15485//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15485//console This message is automatically generated.
    label: code-design
21. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12754878/14370-wait-nofity-v2.txt against master branch at commit 27d3ab43efeabe2a0e1173858b6994b17b5c355b. ATTACHMENT ID: 12754878 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {color:red}-1

findbugs{color}. The patch appears to cause Findbugs (version 2.0.3) to fail. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15501//testReport/ Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15501//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15501//console This message is automatically generated.

22. Ran 14370-wait-nofity-v2.txt locally. Except for TestMasterFailoverWithProcedures , the other tests passed. TestMasterFailoverWithProcedures passed when run alone.
23. Sorry, I have a hard time understanding the reasoning to go back to approach v1 from v3 patch. The possible race conditions I mention above are not specific to the Thread vs Executors, it is orthogonal to that. So v1 patch or a wait-signal version does not buy us anything compared to v3 patch. In the v3 patch, you are submitting a Runnable thread to the executor which runs indefinitely everytime node data changes. The lifecycle of ZKPermissionWatcher is different than AcccessController. I think what happens is that the AcccessController coprocessor will be stopped everytime a region is closed from the region server, while the ZKPermissionWatcher is cached via TableAuthManager. Let me attach a patch, to explain it better.
24. **body:** Ted, can you check the v4 patch to see my points above. I did not run any tests on it though.
    label: test
25. **body:** {code} 778 if (ref-1 == 0) { 779 instance.close(); {code} instance should be removed from refCount, right ? {code} 68 new DaemonThreadFactory("zkpermissionwatcher")); {code} Adding hyphen would make the thread name more readable. Submitting all actions to executor maintains the semantics of original implementation. If I cannot convince you that patch v1 doesn't introduce client visible difference in terms of ACL, we can go with patch v4. However, in a cluster with many tables where new tables are being created, the backing queue for the executor may have non-trivial length.
    label: code-design
26. Patch v5 adds call to watcher.abort() in TableAuthManager#release() when ref counting assertion fails.
27. {color:green}+1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755024/hbase-14370_v4.patch against master branch at commit a11bb2a933ad799546e7179fdf6ce75e3e22d44b. ATTACHMENT ID: 12755024 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 6 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:green}+1 core tests{color}. The patch passed unit tests in . Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15516//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15516//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15516//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15516//console This message is automatically generated.
28. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755052/14370-v5.txt against master branch at commit a11bb2a933ad799546e7179fdf6ce75e3e22d44b. ATTACHMENT ID: 12755052 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 6 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15517//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15517//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15517//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15517//console This message is automatically generated.
29. Patch v7 adds preemption for nodeChildrenChanged events on top of patch v5.
30. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755121/14370-v7.txt against master branch at commit e770cf34174c8226eaf703c303ee3d8397c38242. ATTACHMENT ID: 12755121 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 9 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:red}-1 site{color}. The patch appears to cause mvn post-site goal to fail. {color:red}-1 core tests{color}. The patch failed these unit tests: org.apache.hadoop.hbase.mapreduce.TestImportExport org.apache.hadoop.hbase.util.TestProcessBasedCluster Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15525//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15525//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15525//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15525//console This message is automatically generated.
31. The above test failures were not related to the patch - I have seen them in other QA runs. The tests pass locally.
32. **body:** It is hard to follow what is going on and I don't see a test that exercises this new complexity: i.e. processing loads of znodes asserting it is Doing The Right Thing. This patch includes refcounting without test at extremities (just does log warn that "Something wrong with the TableAuthManager reference counting: "...) Patch passes to executor anonymous new Runnable() throughout but then declares a ' private Runnable runnable = new Runnable() {'... Mainly wanted to say we don't need more threads but you fellas seem to be trying hard to avoid long-running thread that does nothing 99.9999999999999999% of the time so that is good.
    label: code-design
33. **body:** bq. just does log warn that "Something wrong with the TableAuthManager The following change in the same if block would take action on top of warning: {code} + instance.getZKPermissionWatcher().getWatcher().abort(msg, null); {code} bq. but then declares a ' private Runnable The private runnable allows subsequent nodeChildrenChanged event to preempt current processing of previous nodeChildrenChanged event. The rationale is that there is no need to continue processing potentially stale data. Would renaming the private runnable (e.g. nodeChildrenChangedRunnable) make the code more readable ? As of patch v7, the order of handling zk notifications is strictly the same as current formation. As stated earlier, the customer's use case constantly creates new tables. As of last Friday, there were ~2600 tables. I wouldn't be surprised if the table count reaches 3000. Efficiently handling zk notifications becomes important such that the notifications for region assignment are not blocked by the handling for ACL.
    label: code-design
34. **body:** Patch v8 moves the private runnable inside nodeChildrenChanged() with some comment explaining what it does. Hopefully this makes code more readble.
    label: code-design

35. A WARN then it aborts? Tests?
36. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755178/14370-v8.txt against master branch at commit 0f0cdc5131913e1d82e9099c0a8e000c2ac97754. ATTACHMENT ID: 12755178 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 9 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {color:red}-1 findbugs{color}. The patch appears to cause Findbugs (version 2.0.3) to fail. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:red}-1 site{color}. The patch appears to cause mvn post-site goal to fail. {color:red}-1 core tests{color}. The patch failed these unit tests: Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15531//testReport/ Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15531//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15531//console This message is automatically generated.
37. {code} /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/test-framework/dev-support/test-patch.sh: line 861: mvn: command not found We're ok: there is no zombie test {code} How come the above error happened on H5 as well :-) Let me change the log from warn to fatal and try to come up with a unit test.
38. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755178/14370-v8.txt against master branch at commit 0f0cdc5131913e1d82e9099c0a8e000c2ac97754. ATTACHMENT ID: 12755178 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 9 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: {color:red}-1 core zombie tests{color}. There are 1 zombie test(s): at org.apache.hadoop.hbase.regionserver.TestRegionReplicas.testRefreshStoreFiles(TestRegionReplicas.java:237) at org.apache.camel.component.jetty.HttpEndpointUriEncodingIssueTest.testEndpointUriEncodingIssue(HttpEndpointUriEncodingIssueTest.java:32) Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15530//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15530//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15530//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15530//console This message is automatically generated.
39. Here is my test plan: subclass AccessController and override its stop() method. When the subclass stop() method calls super#stop() twice in a row, verify the abortion. Please let me know if other scenario should be tested.
40. Sounds fine. What about refcounting? Any way a recount decrement can be missed?
41. The scenario I described was for refcounting verification where double decrement leads to region server abortion. w.r.t. missed recount decrement, I haven't found a scenario yet.
42. In patch v10, added test where double ref decrement leads to region server abort. Also verified that after tables are deleted, total reference count comes down to 0 in normal operation.
43. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755242/14370-v10.txt against master branch at commit bf26088d7be4386864148516b151dfb0a858c416. ATTACHMENT ID: 12755242 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 14 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not increase the total number of checkstyle errors {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: {color:red}-1 core zombie tests{color}. There are 5 zombie test(s): at org.apache.hadoop.hbase.io.hfile.TestCacheOnWrite.testStoreFileCacheOnWriteInternals(TestCacheOnWrite.java:274) at org.apache.hadoop.hbase.io.hfile.TestCacheOnWrite.testStoreFileCacheOnWrite(TestCacheOnWrite.java:503) at org.apache.hadoop.hbase.replication.regionserver.TestReplicationWALReaderManager.test(TestReplicationWALReaderManager.java:181) at org.apache.hadoop.hbase.TestAcidGuarantees.testMobGetAtomicity(TestAcidGuarantees.java:392) Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15540//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15540//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15540//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15540//console This message is automatically generated.
44. **body:** bq. Mainly wanted to say we don't need more threads but you fellas seem to be trying hard to avoid long-running thread that does nothing 99.999999999999999% of the time so that is good. The original motivation for this patch was due to HBASE-12635 having left a dynamic cluster with lots of regions with 60K acl definitions. The zk watcher thread will spend 3+ minutes just to do the refresh acls. Even with HBASE-12635 fixed, I think we should follow the practice of forking a thread to process the zk notifications. I did not do the perf analysis, but we have a cluster with 2000 tables which may make the refreshNodes() to be in the multi-seconds range. The ref counting is unfortunate, since there is no easy way to have an executor corresponding to a TableAuthManager since TableAuthManager itself is a static cache. We could have added the executor as one of the core RS threads, but that seems also a bit hacky. If there are suggestions there, I can try it out. Coming back to patch, Ted, I think I got the motivation for the preemption. v10 patch looks fine to me.
    label: code-design
45. Rerun failed tests shown above and they passed locally.
46. Patch for branch-1.
47. TestAccessController3 seems to hang in branch-1. Here is part of stack trace: {code} "main" prio=5 tid=0x00007ff522800000 nid=0x1903 in Object.wait() [0x00000001098f7000] java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) - waiting on <0x00000007c6a63358> (a java.util.concurrent.atomic.AtomicBoolean) at org.apache.hadoop.hbase.client.RpcRetryingCaller.callWithRetries(RpcRetryingCaller.java:168) - locked <0x00000007c6a63358> (a java.util.concurrent.atomic.AtomicBoolean) at org.apache.hadoop.hbase.ipc.RegionCoprocessorRpcChannel.callExecService(RegionCoprocessorRpcChannel.java:95) at org.apache.hadoop.hbase.ipc.CoprocessorRpcChannel.callBlockingMethod(CoprocessorRpcChannel.java:73) at org.apache.hadoop.hbase.protobuf.generated.AccessControlProtos$AccessControlService$BlockingStub.grant(AccessControlProtos.java:10280) at org.apache.hadoop.hbase.protobuf.ProtobufUtil.grant(ProtobufUtil.java:2181) at org.apache.hadoop.hbase.security.access.SecureTestUtil$2.call(SecureTestUtil.java:375) at org.apache.hadoop.hbase.security.access.SecureTestUtil$2.call(SecureTestUtil.java:367) at org.apache.hadoop.hbase.security.access.SecureTestUtil.updateACLs(SecureTestUtil.java:332) at org.apache.hadoop.hbase.security.access.SecureTestUtil.grantGlobal(SecureTestUtil.java:367) at org.apache.hadoop.hbase.security.access.TestAccessController3.setUpTableAndUserPermissions(TestAccessController3.java:243) at org.apache.hadoop.hbase.security.access.TestAccessController3.setupBeforeClass(TestAccessController3.java:187) {code}
48. Turns out HBASE-14378 covers TestAccessController* already. Would wait for that to go in first.

49. [~apurtell]: I am preparing backport to 0.98 branch. Let me know if you have comments. Thanks
50. Review looks solid, no comments, please go ahead.
51. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755409/14370-branch-1-v10.txt against branch-1 branch at commit c438052cc2280121727d4ae0883f0b76cad5816a. ATTACHMENT ID: 12755409 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 14 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:red}-1 checkstyle{color}. The applied patch generated 3815 checkstyle errors (more than the master's current 3814 errors). {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: org.apache.hadoop.hbase.snapshot.TestFlushSnapshotFromClient org.apache.hadoop.hbase.mapred.TestTableMapReduce org.apache.hadoop.hbase.client.TestSnapshotCloneIndependence org.apache.hadoop.hbase.mapreduce.TestImportTsv {color:red}-1 core zombie tests{color}. There are 23 zombie test(s): at org.apache.hadoop.hbase.security.visibility.TestVisibilityLabelsWithACL.testVisibilityLabelsForUserWithNoAuths(TestVisibilityLabelsWithACL.java:203) at org.apache.hadoop.hbase.regionserver.TestMajorCompaction.testUserMajorCompactionRequest(TestMajorCompaction.java:429) at org.apache.hadoop.hbase.regionserver.TestSplitTransactionOnCluster.testSplitWithRegionReplicas(TestSplitTransactionOnCluster.java:1024) at org.apache.hadoop.hbase.regionserver.TestCorruptedRegionStoreFile.testLosingFileAfterScannerInit(TestCorruptedRegionStoreFile.java:172) at org.apache.hadoop.hbase.mapred.TestTableSnapshotInputFormat.testInitTableSnapshotMapperJobConfig(TestTableSnapshotInputFormat.java:104) at org.apache.hadoop.hbase.regionserver.TestAtomicOperation.testIncrementMultiThreads(TestAtomicOperation.java:166) at org.apache.hadoop.hbase.snapshot.TestExportSnapshot.testExportFileSystemState(TestExportSnapshot.java:286) at org.apache.hadoop.hbase.snapshot.TestExportSnapshot.testExportFileSystemState(TestExportSnapshot.java:260) at org.apache.hadoop.hbase.snapshot.TestExportSnapshot.testExportFileSystemState(TestExportSnapshot.java:193) at org.apache.hadoop.hbase.mapreduce.TestImportTsv.testMROnTable(TestImportTsv.java:116) at org.apache.hadoop.hbase.mapred.TestTableInputFormat.testInputFormat(TestTableInputFormat.java:353) at org.apache.hadoop.hbase.mapred.TestTableInputFormat.testDeprecatedExtensionOfTableInputFormatBase(TestTableInputFormat.java:334) at org.apache.hadoop.hbase.regionserver.TestRegionReplicas.testRefreshStoreFiles(TestRegionReplicas.java:250) at org.apache.hadoop.hbase.snapshot.TestExportSnapshot.testExportFileSystemState(TestExportSnapshot.java:286) at org.apache.hadoop.hbase.snapshot.TestExportSnapshot.testExportFileSystemState(TestExportSnapshot.java:260) at org.apache.hadoop.hbase.snapshot.TestExportSnapshot.testExportWithTargetName(TestExportSnapshot.java:218) Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15562//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15562//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15562//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15562//console This message is automatically generated.
52. QA environment issue: {code} testRegionCrossingHFileSplitRowBloom(org.apache.hadoop.hbase.mapreduce.TestLoadIncrementalHFilesUseSecurityEndPoint) Time elapsed: 1.308 sec <<< ERROR! org.apache.hadoop.ipc.RemoteException: unable to create new native thread {code} I don't think any of the above test failure is involved with enabling ACL. I am running the tests locally to make sure they pass.
53. Patch for 0.98 branch
54. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12755457/14370-0.98-v10.txt against 0.98 branch at commit c94d10952fe44f73096027cc9083cee993983940. ATTACHMENT ID: 12755457 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 14 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:red}-1 javadoc{color}. The javadoc tool appears to have generated 22 warning messages. {color:red}-1 checkstyle{color}. The applied patch generated 3869 checkstyle errors (more than the master's current 3868 errors). {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15567//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15567//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15567//artifact/patchprocess/checkstyle-aggregate.html Javadoc warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15567//artifact/patchprocess/patchJavadocWarnings.txt Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15567//console This message is automatically generated.
    label: code-design
55. For 0.98 QA run: {code} fht https://builds.apache.org/job/PreCommit-HBASE-Build/15567/console Fetching the console output from the URL Printing hanging tests Hanging test : org.apache.hadoop.hbase.mapreduce.TestTableInputFormatScan1 Hanging test : org.apache.hadoop.hbase.mapreduce.TestMultiTableInputFormat Printing Failing tests {code} The above hanging tests were not related to the patch. Planning to commit in the near future.
56. FAILURE: Integrated in HBase-TRUNK #6800 (See [https://builds.apache.org/job/HBase-TRUNK/6800/]) HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes() (tedyu: rev dff5243c89544de8ed3127a7df5ec79cdab3373b) * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestTablePermissions.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestZKPermissionsWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/AccessController.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/ZKPermissionWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController3.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/SecureTestUtil.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/TableAuthManager.java
57. FAILURE: Integrated in HBase-0.98-on-Hadoop-1.1 #1073 (See [https://builds.apache.org/job/HBase-0.98-on-Hadoop-1.1/1073/]) HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes() (tedyu: rev c5f3f68339f4a87bfa7823a8b8edad33a51b205c) * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController3.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestTablePermissions.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/AccessController.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/TableAuthManager.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestZKPermissionsWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/SecureTestUtil.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/ZKPermissionWatcher.java
58. FAILURE: Integrated in HBase-0.98 #1120 (See [https://builds.apache.org/job/HBase-0.98/1120/]) HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes() (tedyu: rev c5f3f68339f4a87bfa7823a8b8edad33a51b205c) * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestZKPermissionsWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController.java * hbase-

server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController3.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/ZKPermissionWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestTablePermissions.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/SecureTestUtil.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/TableAuthManager.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/AccessController.java

59. Waiting for green branch-1.x build before integrating to those branches.
60. bq.private static Map<TableAuthManager, Integer> refCount = new HashMap<>(); Can't we have refCount info within TableAuthManager instance? The 'release' method can be on instance rather than a static? {quote} public void close() { 107 executor.shutdown(); 108 } {quote} Do we need call shutdownNow() so that the running Runnable may be interrupted ?
61. bq. Can't we have refCount info within TableAuthManager instance Having refCount as static makes tracking outstanding TableAuthManager instance(s) easy. bq. Do we need call shutdownNow() Calling shutdown() allows outstanding notifications to be processed.
62. You want it to be processed? After the close()? bq.Having refCount as static makes tracking outstanding TableAuthManager instance(s) easy But do we have a tracking mechanism?
63. I see this is already committed and only branch-1 commit was pending. Sorry for the delay in review. These are not so serious. Feel free to ignore as of now. If we want we can do later.
64. This has been committed, the issue should be resolved as Fixed.
65. Update fix versions when committed to other branches
66. Re-attaching for test run.
67. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12762313/14370-branch-1-v10.txt against branch-1 branch at commit a33adf2f0b050e9cf9330fd5ab7e200a7dd27d6d. ATTACHMENT ID: 12762313 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 14 new or modified tests. {color:green}+1 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:red}-1 checkstyle{color}. The applied patch generated 3777 checkstyle errors (more than the master's current 3776 errors). {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:green}+1 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: {color:red}-1 core zombie tests{color}. There are 1 zombie test(s): Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/15739//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/15739//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/15739//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/15739//console This message is automatically generated.
68. {code} fht https://builds.apache.org/job/PreCommit-HBASE-Build/15739/consoleFull Fetching the console output from the URL Printing hanging tests Hanging test : org.apache.hadoop.hbase.security.access.TestWithDisabledAuthorization Printing Failing tests {code} From https://builds.apache.org/job/HBase-1.3/jdk=latest1.7,label=Hadoop/203/consoleFull : {code} "main" prio=10 tid=0x00007fa16c00a800 nid=0x4e8 waiting on condition [0x00007fa173269000] java.lang.Thread.State: TIMED_WAITING (sleeping) at java.lang.Thread.sleep(Native Method) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.locateRegionInMeta(ConnectionManager.java:1339) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.locateRegion(ConnectionManager.java:1159) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.relocateRegion(ConnectionManager.java:1130) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.relocateRegion(ConnectionManager.java:1114) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.getRegionLocation(ConnectionManager.java:935) at org.apache.hadoop.hbase.client.HRegionLocator.getRegionLocation(HRegionLocator.java:83) at org.apache.hadoop.hbase.client.RegionServerCallable.prepare(RegionServerCallable.java:79) at org.apache.hadoop.hbase.client.RpcRetryingCaller.callWithRetries(RpcRetryingCaller.java:124) at org.apache.hadoop.hbase.ipc.RegionCoprocessorRpcChannel.callExecService(RegionCoprocessorRpcChannel.java:95) at org.apache.hadoop.hbase.ipc.CoprocessorRpcChannel.callBlockingMethod(CoprocessorRpcChannel.java:73) at org.apache.hadoop.hbase.protobuf.generated.AccessControlProtos$AccessControlService$BlockingStub.grant(AccessControlProtos.java:10280) at org.apache.hadoop.hbase.protobuf.ProtobufUtil.grant(ProtobufUtil.java:2209) at org.apache.hadoop.hbase.security.access.SecureTestUtil$8.call(SecureTestUtil.java:502) at org.apache.hadoop.hbase.security.access.SecureTestUtil$8.call(SecureTestUtil.java:494) at org.apache.hadoop.hbase.security.access.SecureTestUtil.updateACLs(SecureTestUtil.java:324) at org.apache.hadoop.hbase.security.access.SecureTestUtil.grantOnTable(SecureTestUtil.java:494) at org.apache.hadoop.hbase.security.access.TestWithDisabledAuthorization.setUp(TestWithDisabledAuthorization.java:203) {code} So it was not regression.
69. FAILURE: Integrated in HBase-1.3 #204 (See [https://builds.apache.org/job/HBase-1.3/204/]) HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes() (tedyu: rev 52188c5c4a55483bea229c77ab37b9bcbe9b3623) * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController3.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/ZKPermissionWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestZKPermissionsWatcher.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/TableAuthManager.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/AccessController.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/SecureTestUtil.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestTablePermissions.java
70. SUCCESS: Integrated in HBase-1.3-IT #182 (See [https://builds.apache.org/job/HBase-1.3-IT/182/]) HBASE-14370 Use separate thread for calling ZKPermissionWatcher#refreshNodes() (tedyu: rev 52188c5c4a55483bea229c77ab37b9bcbe9b3623) * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/ZKPermissionWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/AccessController.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestTablePermissions.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/SecureTestUtil.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/access/TableAuthManager.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestZKPermissionsWatcher.java * hbase-server/src/test/java/org/apache/hadoop/hbase/security/access/TestAccessController3.java
71. Why is this open again? Commit, then resolve. Don't commit and not resolve. Thanks for helping RMs keep JIRA clean!