Item 168

git_comments:

git_commits:

1. **summary:** HBASE-4217 addendum to change duplicate table name
   message: HBASE-4217 addendum to change duplicate table name git-svn-id: https://svn.apache.org/repos/asf/hbase/trunk@1162291 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
   label: documentation
2. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
3. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
   label: documentation
4. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it

from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.

5. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
6. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
7. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
   label: code-design
8. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
9. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
   description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
   label: code-design
10. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020,

key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
label: code-design

11. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
12. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
13. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
    label: documentation
14. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
    label: code-design
15. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion

accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.

16. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
17. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
18. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.
19. **summary:** HRS.closeRegion should be able to close regions with only the encoded name
    description: We had some sort of an outage this morning due to a few racks losing power, and some regions were left in the following state: ERROR: Region UNKNOWN_REGION on sv4r17s9:60020, key=e32bbe1f48c9b3633c557dc0291b90a3, not on HDFS or in META but deployed on sv4r17s9:60020 That region was deleted by the master but the region server never got the memo. Right now there's no way to force close it because HRS.closeRegion requires an HRI and the only way to create one is to get it from .META. which in our case doesn't contain a row for that region. Basically we have to wait until that server is dead to get rid of the region and make hbck happy. The required change is to have closeRegion accept an encoded name in both HBA (when the RS address is provided) and HRS since it's able to find it anyways from it's list of live regions.

jira_issues_comments:

1. **body:** {code} + public void testShouldNotThrowExceptionIfTheRegionNamePassedToCloseIsNotAValidRegion()^M {code} The method name is informative but too long. How about testShouldNotThrowIfTheRegionNamePassedToCloseIsNotAValidRegion ? Please add javadoc for the new test. {code} +^M +public class TestHBACloseRegion {^M {code}
   label: documentation
2. Thanks Ted for your quick review. Some problem in my editor. Today's patches have that ^M characters. Will see that this does not repeat
3. **body:** A few comments Ram. The method you add to HBaseAdmin needs a different name and the javadoc needs to better specify that what we are expecting is the encoded region name. So something like closeRegionWithEncodedRegionName. Then the javadoc for regionname should explain what the encoded part of a region name is. You need to better differentiate this method from the other two closeRegion methods that are in HBaseAdmin. Similar for the new method added to the HRegionInterface (and you don't need javadoc in HRS since its implementing the Interface so it shoudl

inherit the Interface doc) Do we not have a TestAdmin class already? Should these new tests go in there instead or into class of their own? Otherwise, patch looks great Ram... Oh, what about updating the shell so when you do close_region, it uses your new method? Good stuff.
label: documentation

4. @Stack Thanks for your comments. {bq} javadoc needs to better specify that what we are expecting is the encoded region name The new api is not using the encoded region name. It accepts the region name and servername. If server name is null it tries to invoke the existing closeRegion api. in The exisiting closeRegion {code} Pair<HRegionInfo, ServerName> pair = MetaReader.getRegion(ct, regionname); {code} If we pass the encoded region name here it is not able to retrieve the pair object. Hence i opted to get the actual region name convert to bytes, based on the servername determine which closeRegion to call.Hence the name of the new method was closeRegionByRegionName(). Pls provide your suggestions.
5. @Stack bq. Similar for the new method added to the HRegionInterface (and you don't need javadoc in HRS since its implementing the Interface so it shoudl inherit the Interface doc) {noformat} + @Override + @QosPriority(priority=HIGH_QOS) + public boolean closeRegion(byte[] regionName, boolean zk) throws IOException { {noformat} As in other closeRegion() the javadoc is getting inherited here also. bq. javadoc needs to better specify that what we are expecting is the encoded region name I have done one small modification from the previous patch as per Stack's suggestion. The new api in HBaseAdmin will take the encodedRegion name along with server name. If servername is not given then the api will just log and will not do any. In the previous patch as I had accepted the normal region name if servername was null was calling the older closeRegion() api. bq.Do we not have a TestAdmin class already? Should these new tests go in there instead or into class of their own? I have added the new testcases to TestAdmin bq.Oh, what about updating the shell so when you do close_region, it uses your new method? I have done this now. Yesterday I missed this part. Pls check this if my way of adding the new api in shell is correct. If not kindly correct me so that i can update it.
6. I wrote the below to go against your first comment but your second comment seems to address the below so ignore it (I'm posting it in case the below has not been addressed). I'm going to look at the patch not. Good on you Ram. bq. The new api is not using the encoded region name. This issue is about doing a closeRegion with the encoded name. The HREgionServer has a Map of regions keyed by the encoded name of the region. Its needed for the case described at the head of this issue where we don't have the region name to hand, all we have is the encoded name. The MetaReader.getRegion is not needed if you have the ServerName and the encoded name? You'd use the ServerName to figure the connection to the HRI, then you would invoke closeRegion (In the guts of closeRegion, it could figure it was an encoded name only or if it was a full region name just be looking at what it was passed -- so perhaps you don't even need to add a new method to HRegionInterface, etc., just add a check to the head of the current closeRegion method?
7. **body:** This patch is looking good. I'd say just throw exception here Ram, an NPE which has a message saying servername can't be null or better an illegalargumentexception (you are explicit in your javadoc that this can't be null): {code} + if (null == serverName) { + LOG.error("The server name cannot be null or empty."); + return; + } {code} So you add new method to the HRegionInterface. Thats good. Is the 'regionName' param the encoded name or the HRI#getRegionName as bytes? I think it the former? If so, we should say that in the java doc and change the name of the param to be encodedName rather than regionName. We'd do it in Interface and in HRS. In your closeRegion implementation, seems like you could save some of the work this method is doing by instead calling the hard to find method getFromOnlineRegions. In the shell messing, do we have to add the verb + close_region_by_encoded_regionname? Its kinda long? Can't we just use current closeRegion and then by looking at what is passed, figure which method on HBA to call? Otherwise, this is excellent Ram.
label: code-design
8. Addressed the review comments. Reg the ruby code i have changed it after reading some materials in ruby. Correctness needs to be verified.
9. **body:** isRegionClosed is a bad name for a variable. It is what you would call the method that returns this variable. I'll change it to regionClosed on commit. Patch is great. Going to apply.
label: code-design
10. **body:** This is what I applied. Small cleanup around help text in shell.
label: code-design
11. Applied to TRUNK. Nice patch Ram. Thanks. A patch for 0.90 would be nice since this is pretty important fixup operation -- being able to close region.... but it'd take a bit of work to pull it around.
12. If a 0.90 version, we maybe should do that in another issue.
13. **body:** @Stack Thanks a lot for taking the pains to correct the patch and then commit. I will improve so that your work is readuced. :)

label: documentation

14. Integrated in HBase-TRUNK #2147 (See [https://builds.apache.org/job/HBase-TRUNK/2147/]) HBASE-4217 HRS.closeRegion should be able to close regions with only the encoded name HBASE-4217 HRS.closeRegion should be able to close regions with only the encoded name stack : Files : * /hbase/trunk/CHANGES.txt stack : Files : * /hbase/trunk/CHANGES.txt * /hbase/trunk/src/main/java/org/apache/hadoop/hbase/client/HBaseAdmin.java * /hbase/trunk/src/main/java/org/apache/hadoop/hbase/ipc/HRegionInterface.java * /hbase/trunk/src/main/java/org/apache/hadoop/hbase/regionserver/HRegionServer.java * /hbase/trunk/src/main/ruby/hbase/admin.rb * /hbase/trunk/src/main/ruby/shell/commands/close_region.rb * /hbase/trunk/src/test/java/org/apache/hadoop/hbase/client/TestAdmin.java
15. **body:** Table TestHBACloseRegion3 is used in two unit tests and it is not dropped in between. Here is the addendum I committed: {code} Index: src/test/java/org/apache/hadoop/hbase/client/TestAdmin.java === --- src/test/java/org/apache/hadoop/hbase/client/TestAdmin.java (revision 1162289) +++ src/test/java/org/apache/hadoop/hbase/client/TestAdmin.java (working copy) @@ -1025,7 +1025,7 @@ @Test public void testCloseRegionWhenServerNameIsEmpty() throws Exception { - byte[] TABLENAME = Bytes.toBytes("TestHBACloseRegion3"); + byte[] TABLENAME = Bytes.toBytes("TestHBACloseRegionWhenServerNameIsEmpty"); HBaseAdmin admin = createTable(TABLENAME); HRegionServer rs = TEST_UTIL.getRSForFirstRegionInTable(TABLENAME); @@ -1035,7 +1035,7 @@ for (HRegionInfo regionInfo : onlineRegions) { if (!regionInfo.isMetaRegion() && !regionInfo.isRootRegion()) { if (regionInfo.getRegionNameAsString() - .contains("TestHBACloseRegion3")) { + .contains("TestHBACloseRegionWhenServerNameIsEmpty")) { admin.closeRegionWithEncodedRegionName(regionInfo.getEncodedName(), " "); } {code} I think naming the table in accordance with the test case is a good practice.
    label: code-design
16. +1 Good on one Ted.
17. Integrated in HBase-TRUNK #2151 (See [https://builds.apache.org/job/HBase-TRUNK/2151/]) HBASE-4217 addendum to change duplicate table name tedyu : Files : * /hbase/trunk/src/test/java/org/apache/hadoop/hbase/client/TestAdmin.java
18. @Ted Thanks for committing the addendum. Good on you.
19. This issue was closed as part of a bulk closing operation on 2015-11-20. All issues that have been resolved and where all fixVersions have been released have been closed (following discussions on the mailing list).