Item 11
git_comments:

1. ensure the filter instance is not null
2. * Copyright The Apache Software Foundation * * 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.
3. * * This is a Filter wrapper class which is used in the server side. Some filter * related hooks can be defined in this wrapper. The only way to create a * FilterWrapper instance is passing a client side Filter instance through * {@link org.apache.hadoop.hbase.client.Scan#getFilter()}. *
4. To fix HBASE-6429, Filter with filterRow() returning true is incompatible with scan with limit 1. hasFilterRow() returns true, if either filterRow() or filterRow(kvs) is implemented. 2. filterRow() is merged with filterRow(kvs), so that to make all those row related filtering stuff in the same function.
5. * * Test if Filter is incompatible with scan-limits
6. add filter after batch defined
7. * Copyright The Apache Software Foundation * * 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.
8. row1 => <f1:c1, 1_c1>, <f1:c2, 1_c2>, <f1:c3, 1_c3>, <f1:c4,1_c4>, <f1:c5, 1_c5> row2 => <f1:c1, 2_c1>, <f1:c2, 2_c2>, <f1:c3, 2_c3>, <f1:c4,2_c4>, <f1:c5, 2_c5>
9. no correct result is expected
10. set batch number as 2, which means each Result should contain 2 KVs at most
11. Expect to get following row row2 => <f1:c1, 2_c1>, <f1:c2, 2_c2>, row2 => <f1:c3, 2_c3>, <f1:c4, 2_c4>, row2 => <f1:c5, 2_c5>
12. * Copyright The Apache Software Foundation * * 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.
13. no correct result is expected
14. row1 => <f1:c1, 1_c1, ts=1>, <f1:c2, 1_c2, ts=2>, <f1:c3, 1_c3,ts=3>, <f1:c4,1_c4, ts=4>, <f1:c5, 1_c5, ts=5> row2 => <f1:c1, 2_c1, ts=2>, <f1,c2, 2_c2, ts=2>, <f1:c3, 2_c3,ts=2>, <f1:c4,2_c4, ts=2>, <f1:c5, 2_c5, ts=2> row3 => <f1:c1, 3_c1, ts=3>, <f1:c2, 3_c2, ts=3>, <f1:c3, 3_c3,ts=2>, <f1:c4,3_c4, ts=3>, <f1:c5, 3_c5, ts=3>
15. * * Test if the FilterWrapper retains the same semantics defined in the * {@link org.apache.hadoop.hbase.filter.Filter}
16. row2 (c1-c4) and row3(c1-c4) are returned

git_commits:

1. **summary:** HBASE-6429 Filter with filterRow() returning true is incompatible with scan with limit (Jie Huang)

message: HBASE-6429 Filter with filterRow() returning true is incompatible with scan with limit (Jie Huang) git-svn-id: https://svn.apache.org/repos/asf/hbase/trunk@1368472 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Filter with filterRow() returning true is incompatible with scan with limit
   description: Currently if we scan with bot limit and a Filter with filterRow(List<KeyValue>) implemented, an IncompatibleFilterException will be thrown. The same exception should also be thrown if the filer has its filterRow() implemented.

jira_issues_comments:

1. Assume we have a table as follows: {noformat} row1 => <f1:c1, 1_c1> , <f1:c2 , 1_c2>, <f1:c3 , 1_c3>, <f1:c4 , 1_c4>, <f1:c5 , 1_c5> row2 => <f1:c1 , 2_c1> , <f1:c2 , 2_c2>, <f1:c3 , 2_c3>, <f1:c4 , 2_c4>, <f1:c5 , 2_c5> {noformat} And we scan the table using the Scan below: {noformat} Scan scan = new Scan(); scan.setBatch(2); SingleColumnValueFilter filter = new SingleColumnValueFilter(Bytes.toBytes("f1"), Bytes.toBytes("c5"), CompareFilter.CompareOp.EQUAL, new SubstringComparator("2_c5")); scan.setFilter(filter); {noformat} The results that we got are: {noformat} <f1:c1, 1_c1> , <f1:c2 , 1_c2>, <f1:c3 , 1_c3>, <f1:c4 , 1_c4>, <f1:c1 , 2_c1> , <f1:c2 , 2_c2>, <f1:c3 , 2_c3>, <f1:c4 , 2_c4>, <f1:c5 , 2_c5> {noformat} which is not correct. Since we cannot ensure that the column related to the filter criteria can be scanned with limit, an IncompatibleFilterException should be thrown here.
2. **body:** Checking the 0.94.0 version, it has the similar problem as well. See the nextInternal() function in HRegion.java file. {noformat} do { this.storeHeap.next(results, limit - results.size(), metric); if (limit > 0 && results.size() == limit) { if (this.filter != null && filter.hasFilterRow()) { <<< check the compatibility between filterRow(kvs) and scan with limit only throw new IncompatibleFilterException("Filter with filterRow(List<KeyValue>) incompatible with scan with limit!"); } return true; // we are expecting more yes, but also limited to how many we can return. } } while (Bytes.equals(currentRow, nextRow = peekRow())); {noformat} According to current implementation, "hasFilterRow()" function is the only way to invoke "filterRow(kvs)", which means that filter has some extra filtering work on the entire row. Furthermore, the user has another chance to clear that entire row only if "boolean filterRow()" returns true. The semantics of those 2 functions , hasFilterRow() and boolean filterRow(), are quite similar. From my perspective, it would be better to merge two criterion into one. a) We can simply make hasFilterRow() returning true, if either filterRow() or filterRow(kvs) has been implemented. b) To do the result.clear() in the filterRow(kvs) function if filterRow() returns true. It not only fixes this issue, but also somehow eliminates some confusion. If someone wants to do something on the entire row (like clear all or parts of the row content) in filter, just to return true in hasFilterRow() in their Filter implementation. And at them same time, the cleaning job will be called in the filterRow(kvs) function by checking the returned value of "boolean filterRow()", which means all those changings on a row happen in filterRow(kvs). Any comment? OR I can try to upload one patch later.
   label: code-design
3. bq. if either filterRow() or filterRow(kvs) has been implemented. Please clarify the above: the two methods are defined in FilterBase and Filter. Filters have to implement them.
4. Yes. Filter is an interface to define those necessary APIs in "FilterImpl". And FilteBase just "implements" those APIs with default or blank function block. If the actual FilterImpl (like SingleColumnValueFiler) overrides or really needs those 2 functions, it means some modifications are expected on the entire row. Consequently, we should make hasFilterRow() to be ture.
5. 1. Add one FilterWrapper to call "if (filterRow()) kvs.clear()" at the end of filterRow(kvs) function. 2. Make hasFilterRow() return true while some modifications are expected in Filter, such as clearing or

updating the row content.

6. {quote} Please clarify the above: the two methods are defined in FilterBase and Filter. Filters have to implement them. {quote} The default implementations of these two methods in FilterBase do not filter anything. A filter need to provide its own implementation of the function if it wish to filter a row out - that's what is referred to in the description.
7. **body:** @Jie: Have you run unit tests for your patch ? I think certain code may actually rely on the existing behavior to work. We should be careful not to introduce regression.
   label: code-design
8. **body:** Patch seems to be generated from old 0.94 snapshot. When I applied the patch, I had to deal with: src/main/java/org/apache/hadoop/hbase/regionserver/HRegion.java.rej I got two test failures: {code} Failed tests: testPageFilter(org.apache.hadoop.hbase.filter.TestFilter): Expected 6 rows but scanned 4 rows expected:<6> but was:<4> testWhileMatchFilterWithFilterRow(org.apache.hadoop.hbase.filter.TestFilter): The page filter returned more rows than expected expected:<4> but was:<3> {code}
   label: code-design
9. Oops.I will fix those 2 failures and regenerate the patch soon. Thanks Ted.
10. 1. Prepare a patch against trunk 2. Add one more unit test case (TestFilterWithScanLimits) 3. Fix 2 unit test failures in the previous version.
11. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12537300/hbase-6429-trunk.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. +1 hadoop2.0. The patch compiles against the hadoop 2.0 profile. +1 javadoc. The javadoc tool did not generate any warning messages. -1 javac. The applied patch generated 5 javac compiler warnings (more than the trunk's current 4 warnings). -1 findbugs. The patch appears to introduce 12 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed these unit tests: org.apache.hadoop.hbase.TestCheckTestClasses Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop1-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/2419//console This message is automatically generated.
12. TestFilterWithScanLimits.java and FilterWrapper.java need Apache license. {code} + if(null == filter) { {code} Space between if and (. Why does TestFilterWithScanLimits have main() method ? It should be classified as medium test.
13. Thank you very much, Ted. Here updates that trunk patch file. More comment, please let me know.
14. **body:** Year is not needed in license header of FilterWrapper.java and TestFilterWithScanLimits.java {code} + // Wrap the following function calls + @Override + public void filterRow(List<KeyValue> kvs) { {code} Please change the above javadoc so that it explains the rationale behind this JIRA. {code} + if(scan.hasFilter()){ {code} Space between if and (. {code} + "Filter with filterRow(List<KeyValue>) or filterRow() incompatible with scan with limit!"); {code} Line length limit is 100 characters. In the test: {code} + } catch (IOException e) { + // TODO Auto-generated catch block + e.printStackTrace(); {code} I think we should fail the test above. There're other TODO's in test code which needs handling. Please run your next patch through unit tests before attaching. Thanks
    label: code-design
15. Thank you very much for your time Ted. Really sorry for my mistakes. All tests are passed with this version according the local results. More comments, please let me know. thanks.
16. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12537537/hbase-6429-trunk.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. +1 hadoop2.0. The patch compiles against the hadoop 2.0 profile. +1 javadoc. The javadoc tool did not generate any warning messages. -1 javac. The applied patch generated

5 javac compiler warnings (more than the trunk's current 4 warnings). -1 findbugs. The patch appears to introduce 14 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed these unit tests: org.apache.hadoop.hbase.catalog.TestMetaReaderEditor Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop1-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/2430//console This message is automatically generated.

17. Re-run TestMetaReaderEditor locally, PASSED. {noformat} -- T E S T S --- Running org.apache.hadoop.hbase.catalog.TestMetaReaderEditor Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 126.53 sec Results : Tests run: 5, Failures: 0, Errors: 0, Skipped: 0 {noformat}
18. For filterRow(List<KeyValue> kvs): {code} + this.filter.filterRow(kvs); + if (!kvs.isEmpty() && this.filter.filterRow()) { + kvs.clear(); + } {code} Can we exchange the order of filterRow(kvs) and if statement ? If this.filter.filterRow() returns true, we don't need to call filterRow(kvs). Correct me if I am wrong.
19. I am afraid we cannot do that. See the following case : {code} // pseudocode void filterRow(List<KeyValue> kvs){ if kvs.contains(target_kv) { kvs.remove(target_kv) find_target_kv_count++ } } // According to the javadoc, it is the Last chance to veto row based on previous filterKeyValue(KeyValue) calls. boolean filterRow() { return true; // kvs is empty finally } Case A void filterRowWrapper(List<KeyValue> kvs) { this.filter.filterRow(kvs) // find_target_kv_count++ has been called here if (!kvs.isEmpty() && this.filter.filterRow()) { kvs.clear(); // kvs is empty here } // find_target_kv_count has been changed } Case B void filterRowWrapper(List<KeyValue> kvs) { if (!kvs.isEmpty() && this.filter.filterRow()) { kvs.clear(); // kvs is empty here } if(!this.filter.filterRow()) { this.filter.filterRow(kvs); } // find_target_kv_count hasn't been changed } {code} Any question , please let me know.
20. Is find_target_kv_count a field of the filter which is wrapped by FilterWrapper ? Where would the latest value of find_target_kv_count be checked ? Thanks
21. This is just a hypothesis. The find_target_kv_count is a user defined variable in real "FilterImpl". The user may need this value to decide if filterRow() returns true or false , like "if (find_target_kv_count > xxx) return true;".
22. List<KeyValue> is passed to filterRow(kvs). This means user code can clear the list when find_target_kv_count > xxx. It seems there is no need to wait till filterRow() is called. Thanks
23. We should keep filterRow() call after the call to filterRow(kvs). This would align with the existing semantics.
24. Here is another example using existing filter. {code} private static void prepareData() { try { HTable table = new HTable(TestFilter.conf, name); assertTrue("Fail to create the table", admin.tableExists(name)); List<Put> puts = new ArrayList<Put>(); // row1 => <f1:c1 , 1_c1, ts=1> , <f1:c2 , 1_c2,ts=2>, <f1:c3 , 1_c3,ts=3>, <f1:c4 , // 1_c4,ts=4>, <f1:c5 , 1_c5,ts=5> // row2 => <f1:c1 , 2_c1, ts=2> , <f1:c2 , 2_c2, ts=2>, <f1:c3 , 2_c3, ts=2>, <f1:c4 , // 2_c4, ts=2>, <f1:c5 , 2_c5, ts=2> // row3 => <f1:c1 , 3_c1, ts=3> , <f1:c2 , 3_c2, ts=3>, <f1:c3 , 3_c3, ts=3>, <f1:c4 , // 3_c4, ts=3>, <f1:c5 , 3_c5, ts=3> for (int i = 1; i < 4; i++) { Put put = new Put(Bytes.toBytes("row" + i)); for (int j = 1; j < 6; j++) { long timestamp = j; if(i!=1) timestamp = i; put.add(Bytes.toBytes("f1"), Bytes.toBytes("c" + j), timestamp, Bytes.toBytes(i + "_c" + j)); } puts.add(put); } table.put(puts); table.close(); } catch (IOException e) { e.printStackTrace(); assertNull("Exception found while putting data into table", e); } } public testFilter() { int kv_number = 0; int row_number = 0; try { Scan scan = new Scan(); List<Filter> fs = new ArrayList<Filter>(); DependentColumnFilter f1 = new DependentColumnFilter(Bytes.toBytes("f1"), Bytes.toBytes("c5"), true, CompareFilter.CompareOp.EQUAL, new SubstringComparator("c5")); PageFilter f2 = new PageFilter(2); fs.add(f1); fs.add(f2); FilterList filter = new FilterList(fs); scan.setFilter(filter); HTable table = new HTable(conf, name); ResultScanner scanner = table.getScanner(scan); // row2 (c1-c4) and

row3(c1-c4) are returned for (Result result : scanner) { row_number++; for (KeyValue kv : result.list()) { LOG.debug(kv_number + ". kv: " + kv); kv_number++; assertEquals("Returned row is not correct", new String(kv.getRow()), "row" + (row_number+1)); } } scanner.close(); table.close(); } catch (Exception e) { assertNull("Exception happens in scan", e); } LOG.debug("check the fetched kv number"); assertEquals("We should get 8 results returned.", 8, kv_number); assertEquals("We should get 2 rows returned", 2, row_number); } {code}

25. The above case can show the difference. If it is necessary, we can add the test case later.
26. Thanks Ted. I have added that new test case in TestFilterWrapper file, which checks if the FilterWrapper keeps the right semantics.
27. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12537801/hbase-6429-trunk-v2.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. +1 hadoop2.0. The patch compiles against the hadoop 2.0 profile. +1 javadoc. The javadoc tool did not generate any warning messages. -1 javac. The applied patch generated 5 javac compiler warnings (more than the trunk's current 4 warnings). -1 findbugs. The patch appears to introduce 14 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed these unit tests: org.apache.hadoop.hbase.TestRegionRebalancing org.apache.hadoop.hbase.master.TestSplitLogManager Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop1-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/2435//console This message is automatically generated.
28. Will go over patch v2 one more time. Feedback from other people is welcome.
29. **body:** {code} + * This is a Filter wrapper class which is used in the server side. Some filter + * related hooks can be defined in this wrapper. The only way to create a + * FilterWrapper instance is passing a client side Filter instance. {code} To help users make sense of the third sentence, explanation should be added as to how client Filter instance is obtained by server (through Scan object). {code} + throw new NullPointerException("Cannot create FilterWrapper with null"); {code} 'with null' -> 'with null Filter' {code} + //2. Here to merge the filterRow() into filterRow(kvs), {code} Rephrase the above: filterRow() is merged with filterRow(kvs) {code} - private Filter filter; + private FilterWrapper filter; {code} There is no need to change the type of filter - FilterWrapper doesn't add new methods. {code} + "Filter hasFilterRow incompatible with scan with limit!"); {code} Rephrase the above: 'Filter whose hasFilterRow() returns true is ' {code} + } else { + LOG.error("HBaseAdmin is not initialized successfully."); + } {code} The above condition should result in test failure, right ? {code} + } catch (IOException e) { + e.printStackTrace(); + assertNull("Exception found deleting the table", e); + } {code} Since exception is part of the assertion, e.printStackTrace() is redundant. Please remove the call. {code} + * To test if the FilterWrapper keeps the same semantics {code} The above sentence seems incomplete. {code} + // no correct result is expected + LOG.info("Exception is catched"); + assertNull("Exception happens in scan", e); {code} Spelling mistake in the log statement. Since exception is part of assertion, the log is redundant. Suggest removing the log statement. {code} + } catch (IOException e) { + e.printStackTrace(); + assertNull("Exception found deleting the table", e); + } {code} Please remove e.printStackTrace() call.
    label: code-design
30. Thanks Ted. Here attaches the updated version.
31. **body:** -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12538293/hbase-6429-trunk-v3.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. +1 hadoop2.0. The patch compiles against the hadoop 2.0 profile. +1 javadoc. The javadoc tool did not generate any warning messages. -1 javac. The applied patch generated 5 javac compiler warnings (more than the trunk's current 4 warnings). -1 findbugs. The patch appears to

introduce 5 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed these unit tests: org.apache.hadoop.hbase.client.TestFromClientSide Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop1-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/2453//console This message is automatically generated.
label: code-design

32. TestFromClientSide is passed on my local server. {noformat} --- T E S T S --- Running org.apache.hadoop.hbase.client.TestFromClientSide Tests run: 56, Failures: 0, Errors: 0, Skipped: 3, Time elapsed: 170.132 sec Results : Tests run: 56, Failures: 0, Errors: 0, Skipped: 3 {noformat}
33. Almost there. {code} +public class FilterWrapper implements Filter { {code} Suggest adding the following to above class: {code} @InterfaceAudience.Private @InterfaceStability.Evolving {code}
34. Thanks. Quite reasonable. Done in the updated version (v4).
35. **body:** -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12538863/hbase-6429-trunk-v4.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. +1 hadoop2.0. The patch compiles against the hadoop 2.0 profile. +1 javadoc. The javadoc tool did not generate any warning messages. -1 javac. The applied patch generated 5 javac compiler warnings (more than the trunk's current 4 warnings). -1 findbugs. The patch appears to introduce 6 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed these unit tests: org.apache.hadoop.hbase.client.TestFromClientSide Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop1-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/2478//console This message is automatically generated.
    label: code-design
36. TestFromClientSide#testPoolBehavior is not related to Filter's. I ran the test individually and it passed. Will integrate tomorrow if there is no objection.
37. Integrated to trunk. Thanks for the patch, Jie.
38. Integrated in HBase-TRUNK #3189 (See [https://builds.apache.org/job/HBase-TRUNK/3189/]) HBASE-6429 Filter with filterRow() returning true is incompatible with scan with limit (Jie Huang) (Revision 1368472) Result = SUCCESS tedyu : Files : * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/Filter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/FilterWrapper.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/PageFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/RandomRowFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/SingleColumnValueFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/SkipFilter.java * /hbase/trunk/hbase-

server/src/main/java/org/apache/hadoop/hbase/filter/WhileMatchFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/HRegion.java * /hbase/trunk/hbase-server/src/test/java/org/apache/hadoop/hbase/filter/TestFilterWithScanLimits.java * /hbase/trunk/hbase-server/src/test/java/org/apache/hadoop/hbase/filter/TestFilterWrapper.java

39. Integrated in HBase-TRUNK-on-Hadoop-2.0.0 #119 (See [https://builds.apache.org/job/HBase-TRUNK-on-Hadoop-2.0.0/119/]) HBASE-6429 Filter with filterRow() returning true is incompatible with scan with limit (Jie Huang) (Revision 1368472) Result = FAILURE tedyu : Files : * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/Filter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/FilterWrapper.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/PageFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/RandomRowFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/SingleColumnValueFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/SkipFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/filter/WhileMatchFilter.java * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/HRegion.java * /hbase/trunk/hbase-server/src/test/java/org/apache/hadoop/hbase/filter/TestFilterWithScanLimits.java * /hbase/trunk/hbase-server/src/test/java/org/apache/hadoop/hbase/filter/TestFilterWrapper.java
40. [~zhihyu@ebaysf.com] can we close this out?
41. Marking closed.