Item 30
git_comments:

git_commits:

1. **summary:** HBASE-10148 [VisibilityController] Tolerate regions in recovery
   message: HBASE-10148 [VisibilityController] Tolerate regions in recovery git-svn-id: https://svn.apache.org/repos/asf/hbase/branches/0.98@1552020 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** [VisibilityController] Tolerate regions in recovery
   description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
2. **summary:** [VisibilityController] Tolerate regions in recovery
   description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name:

"org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value:
"org.apache.hadoop.hbase.exceptions.RegionInRecoveryException:
hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at
org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at
org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at
org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at
org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096)
at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)"
{noformat} Should we try to ride over this?

3. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java
========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
4. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java
========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
5. **summary:** [VisibilityController] Tolerate regions in recovery

description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

6. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

7. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes

TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

8. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
9. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at

org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
label: code-design

10. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
11. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
12. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-

common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

13. **summary:** [VisibilityController] Tolerate regions in recovery
    description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
    label: code-design
14. **summary:** [VisibilityController] Tolerate regions in recovery
    description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR

[RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

15. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

16. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096)

at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

17. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
18. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
19. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean

DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

20. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

21. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value:

"org.apache.hadoop.hbase.exceptions.RegionInRecoveryException:
hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at
org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at
org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at
org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at
org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096)
at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)"
{noformat} Should we try to ride over this?

22. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java
========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
23. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java
========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
24. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java

==========
---
hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575)
+++
hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy)
@@ -794,7 +794,7 @@
/** Conf key that enables unflushed WAL edits directly being replayed to region servers */
public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay";
-
public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false;
+
public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true;
public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering";
public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false;
{noformat}
causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering:
{noformat}
2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering
2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)"
{noformat}
Should we try to ride over this?

label: code-design

25. **summary:** [VisibilityController] Tolerate regions in recovery

description: Ted Yu reports that enabling distributed log replay by default, like:
{noformat}
Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java
==========
---
hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575)
+++
hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy)
@@ -794,7 +794,7 @@
/** Conf key that enables unflushed WAL edits directly being replayed to region servers */
public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay";
-
public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false;
+
public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true;
public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering";
public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false;
{noformat}
causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering:
{noformat}
2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering
2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)"
{noformat}
Should we try to ride over this?

26. **summary:** [VisibilityController] Tolerate regions in recovery

description: Ted Yu reports that enabling distributed log replay by default, like:
{noformat}
Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java
==========
---
hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575)
+++
hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy)
@@ -794,7 +794,7 @@
/** Conf key that enables unflushed WAL edits directly being replayed to region servers */
public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay";
-
public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false;
+
public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true;
public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering";
public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false;
{noformat}
causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label

table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

27. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

28. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

29. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?
label: code-design

30. **summary:** [VisibilityController] Tolerate regions in recovery
description: Ted Yu reports that enabling distributed log replay by default, like: {noformat} Index: hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java ========== --- hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (revision 1550575) +++ hbase-common/src/main/java/org/apache/hadoop/hbase/HConstants.java (working copy) @@ -794,7 +794,7 @@ /** Conf key that enables unflushed WAL edits directly being replayed to region servers */ public static final String DISTRIBUTED_LOG_REPLAY_KEY = "hbase.master.distributed.log.replay"; - public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = false; + public static final boolean DEFAULT_DISTRIBUTED_LOG_REPLAY_CONFIG = true; public static final String DISALLOW_WRITES_IN_RECOVERING = "hbase.regionserver.disallow.writes.when.recovering"; public static final boolean DEFAULT_DISALLOW_WRITES_IN_RECOVERING_CONFIG = false; {noformat} causes TestVisibilityController#testAddVisibilityLabelsOnRSRestart to fail. It reveals an issue with label operations if the label table is recovering: {noformat} 2013-12-12 14:53:53,133 DEBUG [RpcServer.handler=2,port=58108] visibility.VisibilityController(1046): Adding the label XYZ2013-12-12 14:53:53,137 ERROR [RpcServer.handler=2,port=58108] visibility.VisibilityController(1074): org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering 2013-12-12 14:53:53,151 DEBUG [main] visibility.TestVisibilityLabels(405): response from addLabels: result { exception { name: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException" value: "org.apache.hadoop.hbase.exceptions.RegionInRecoveryException: hbase:labels,,1386888826648.f14a399ba85cbb42c2c3b7547bf17c65. is recovering at org.apache.hadoop.hbase.regionserver.HRegion.startRegionOperation(HRegion.java:5555) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1763) at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:1749) at org.apache.hadoop.hbase.security.visibility.VisibilityController.getExistingLabelsWithAuths(VisibilityController.java:1096) at org.apache.hadoop.hbase.security.visibility.VisibilityController.postBatchMutate(VisibilityController.java:672)" {noformat} Should we try to ride over this?

jira_issues_comments:

1. Riding over the recovery is feasible. HRegion has this method: {code} public boolean isRecovering() { {code}
2. The issue is very clear now. When log replay is in place, postOpen() hook is getting called before any replay. In case of Visibility , we use this hook to read the existing labels and initialize the cache. So here what happens is the cache is

getting no items at all. Also we calculate the next ordinal no# for adding label in this postOpen(). It comes as 1 again. While doing the 2 new entries write, it gets ordinals as 1 and 2 replacing some old labels !!! That is why the test fails with number of labels after adding 2 new labels (already 6 in place) as 6 only. Similar issue is there with AccessController also in which the existing data is cached in postOpen(). If some unflushed acls are in place, then those will be lost to get added into the cache (up on RS restart).. I am trying with possible solutions. Can we think of a hook with replay like postReplay() or so?

3. Yes, a hook like postReplay could work. So we would check in postOpen if the region isn't ready yet and defer to the new hook to finish init. This also means we need to deny access to everything but the superuser until init is finished, but as long as we document this as he price one pays for security this should be ok.
4. Yes exactly these way I am fixing this... Let me give patch with the new hook and some small changes in the replay flow.. I will give it in another Jira and this Jira will change only VisibilityController.. Also Andy will you open a Jira for the corresponding change in the AccessController ?
5. The following change may be needed so that VisibilityController can determine whether to wait for replay to complete: {code} Index: hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/handler/OpenRegionHandler.java === --- hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/handler/OpenRegionHandler.java (revision 1550767) +++ hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/handler/OpenRegionHandler.java (working copy) @@ -143,14 +143,14 @@ } // check if we need set current region in recovering state - region.setRecovering(false); Map<String, HRegion> recoveringRegions = this.rsServices.getRecoveringRegions(); if (recoveringRegions != null && !recoveringRegions.isEmpty() && recoveringRegions.containsKey(region.getRegionInfo().getEncodedName())) { region.setRecovering(true); recoveringRegions.put(region.getRegionInfo().getEncodedName(), region); - } + } else region.setRecovering(false); + boolean failed = true; if (tickleOpening("post_region_open")) { if (updateMeta(region)) { {code}
6. bq. Also Andy will you open a Jira for the corresponding change in the AccessController ? See HBASE-10161
7. Hey [~anoop.hbase] I see now you already opened HBASE-10160. I'm going to close that as a dup since you asked me to open an issue and then went ahead and did it yourself. :-)
8. No problems Andy..I should have been waiting for some more time :) I will attach a fix once we conclude on the new hook and HBASE-10155.
9. **body:** Patch is basically ok. {noformat} + new RegionInRecoveryException("Labels table is recovering"), response); {noformat} We should say something about the VisibilityController not yet being initialized instead. In TestVisibilityLabels, waitForLabelsRegionAvailability could be rewritten using Predicates with timeouts.
   label: code-design
10. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12619008/HBASE-10148.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified tests. {color:green}+1 hadoop1.0{color}. The patch compiles against the hadoop 1.0 profile. {color:green}+1 hadoop1.1{color}. The patch compiles against the hadoop 1.1 profile. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) 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 site goal to fail. {color:green}+1 core tests{color}. The patch passed unit tests in . Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-thrift.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/8194//console This message is automatically generated.
11. Patch looks good to me apart from that log msg as told by Andy.
12. Patch fixing comments
13. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12619042/HBASE-10148_V2.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}.

The patch appears to include 3 new or modified tests. {color:green}+1 hadoop1.0{color}. The patch compiles against the hadoop 1.0 profile. {color:green}+1 hadoop1.1{color}. The patch compiles against the hadoop 1.1 profile. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) 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 site goal to fail. {color:red}-1 core tests{color}. The patch failed these unit tests:
org.apache.hadoop.hbase.replication.TestReplicationSyncUpTool Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-thrift.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/8200//console This message is automatically generated.
label: code-design

14. Don't think failure in TestReplicationSyncUpTool is related to this patch
15. [~ram_krish], [~apurtell] you guys fine with patch V2 ?
16. +1
17. I don't think RegionInRecoveryException is the correct exception to throw. The CP is not initialized yet. There could be other causes besides the region being in recovery, such as an RPC before the CP is initialized. :-)
18. Got ur concern now Andy.. There is CoprocessorException , can we use that?
19. bq. There is CoprocessorException , can we use that Sounds good to me.
20. +1 on V3 patch
21. Woww.. that is super fast... Thanks Andy... Will commit once QA passes with V3 patch.
22. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12619247/HBASE-10148_V3.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified tests. {color:green}+1 hadoop1.0{color}. The patch compiles against the hadoop 1.0 profile. {color:green}+1 hadoop1.1{color}. The patch compiles against the hadoop 1.1 profile. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) 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 site goal to fail. {color:red}-1 core tests{color}. The patch failed these unit tests:
org.apache.hadoop.hbase.security.visibility.TestVisibilityLabels Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-thrift.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-

Build/8208//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/8208//console This message is automatically generated.

23. Attaching again for QA to start
24. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12619263/HBASE-10148_V3.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified tests. {color:green}+1 hadoop1.0{color}. The patch compiles against the hadoop 1.0 profile. {color:green}+1 hadoop1.1{color}. The patch compiles against the hadoop 1.1 profile. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) 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 site goal to fail. {color:red}-1 core tests{color}. The patch failed these unit tests: org.apache.hadoop.hbase.coprocessor.TestRegionServerCoprocessorExceptionWithAbort Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-thrift.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/8210//console This message is automatically generated.
    label: code-design
25. org.apache.hadoop.hbase.coprocessor.TestRegionServerCoprocessorExceptionWithAbort passing locally.
26. Committed to trunk and 0.98. Thanks for the reviews
27. Thanks Anoop.
28. FAILURE: Integrated in HBase-0.98-on-Hadoop-1.1 #18 (See [https://builds.apache.org/job/HBase-0.98-on-Hadoop-1.1/18/]) HBASE-10148 [VisibilityController] Tolerate regions in recovery (anoopsamjohn: rev 1552020) * /hbase/branches/0.98/hbase-server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * /hbase/branches/0.98/hbase-server/src/test/java/org/apache/hadoop/hbase/security/visibility/TestVisibilityLabels.java
29. SUCCESS: Integrated in HBase-0.98 #21 (See [https://builds.apache.org/job/HBase-0.98/21/]) HBASE-10148 [VisibilityController] Tolerate regions in recovery (anoopsamjohn: rev 1552020) * /hbase/branches/0.98/hbase-server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * /hbase/branches/0.98/hbase-server/src/test/java/org/apache/hadoop/hbase/security/visibility/TestVisibilityLabels.java
30. SUCCESS: Integrated in HBase-TRUNK #4736 (See [https://builds.apache.org/job/HBase-TRUNK/4736/]) HBASE-10148 [VisibilityController] Tolerate regions in recovery (anoopsamjohn: rev 1552021) * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * /hbase/trunk/hbase-server/src/test/java/org/apache/hadoop/hbase/security/visibility/TestVisibilityLabels.java
31. SUCCESS: Integrated in HBase-TRUNK-on-Hadoop-1.1 #11 (See [https://builds.apache.org/job/HBase-TRUNK-on-Hadoop-1.1/11/]) HBASE-10148 [VisibilityController] Tolerate regions in recovery (anoopsamjohn: rev 1552021) * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * /hbase/trunk/hbase-server/src/test/java/org/apache/hadoop/hbase/security/visibility/TestVisibilityLabels.java
32. Closing this issue after 0.99.0 release.