Item 36
git_comments:

1. InterruptedException too. If so, we failed. Even if tickle opening fails then it is a failure.

git_commits:

1. **summary:** HBASE-4452 Possibility of RS opening a region though tickleOpening fails due to
   message: HBASE-4452 Possibility of RS opening a region though tickleOpening fails due to znode version mismatch (Ramkrishna) git-svn-id: https://svn.apache.org/repos/asf/hbase/trunk@1174447 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch
   description: Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.
2. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch
   description: Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now;

tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

3. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.
4. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch

description: Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.
label: code-design

5. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch
description: Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG

org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

6. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

7. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG

org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

8. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

9. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356

DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

10. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

11. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5

not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

12. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.
13. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch **description:** Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN

org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

14. **summary:** Possibility of RS opening a region though tickleOpening fails due to znode version mismatch
    description: Consider the following code {code} long period = Math.max(1, assignmentTimeout/ 3); long lastUpdate = now; while (!signaller.get() && t.isAlive() && !this.server.isStopped() && !this.rsServices.isStopping() && (endTime > now)) { long elapsed = now - lastUpdate; if (elapsed > period) { // Only tickle OPENING if postOpenDeployTasks is taking some time. lastUpdate = now; tickleOpening("post_open_deploy"); } {code} Whenever the postopenDeploy tasks takes considerable time we try to tickleOpening so that there is no timeout deducted. But before it could do this if the TimeoutMonitor tries to assign the node to another RS then the other RS will move the node from OFFLINE to OPENING. Hence when the first RS tries to do tickleOpening the operation will fail. Now here lies the problem, {code} String encodedName = this.regionInfo.getEncodedName(); try { this.version = ZKAssign.retransitionNodeOpening(server.getZooKeeper(), this.regionInfo, this.server.getServerName(), this.version); } catch (KeeperException e) { {code} Now this.version becomes -1 as the operation failed. Now as in the first code snippet as the return type is not captured after tickleOpening() fails we go on with moving the node to OPENED. Here again we dont have any check for this condition as already the version has been changed to -1. Hence the OPENING to OPENED becomes successful. Chances of double assignment. {noformat} 2011-09-22 00:57:29,930 WARN org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempt to transition the unassigned node for 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENING failed, the node existed but was version 5 not the expected version 2 2011-09-22 00:57:33,494 WARN org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Failed refreshing OPENING; region=69797d064f773d1aa9adba56e7ff90a3, context=post_open_deploy 2011-09-22 00:58:02,356 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Attempting to transition node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:11,853 DEBUG org.apache.hadoop.hbase.zookeeper.ZKAssign: regionserver:60020-0x1328ceaa1ff000d Successfully transitioned node 69797d064f773d1aa9adba56e7ff90a3 from RS_ZK_REGION_OPENING to RS_ZK_REGION_OPENED 2011-09-22 00:58:13,956 DEBUG org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler: Opened t9,,1316633193606.69797d064f773d1aa9adba56e7ff90a3. {noformat} Correct me if this analysis is wrong.

jira_issues_comments:

1. Good catch, Ramkrishna. This is the only place where return value from tickleOpening() isn't checked.
2. Running the test cases.. will submit the patch tomorrow morning.
3. Ted testcases are running fine with this. Hope it does not cause any problem like HBASE-4153 :)
4. **body:** Minor comment: {code} + // InterruptedException too. If so, we failed. Even if tickle opening fails + // then it is a failure. {code} I think we don't need 'Even' above. Also, I would initialize the new boolean with false. Running test suite.
   label: code-design
5. @Ted, reason for initializing to true is like the tickleOpening may not be invoked always unlesss the PostOpenDeploytask takes more time. That is why i initialized to true. Want to change it ?
6. @Ramkrishna: Your consideration makes sense. There is no need to change.

7. Test suite passed with patch. +1.
8. +1
9. lgtm. nice catch. pulling in to 0.92
10. Patch for 0.90 branch. TestOpenRegionHandler passes.
11. Integrated to 0.90, 0.92 and TRUNK. Thanks for the patch Ramkrishna. Thanks for the review Stack and Jonathan.
12. Integrated in HBase-TRUNK #2244 (See [https://builds.apache.org/job/HBase-TRUNK/2244/]) HBASE-4452 Possibility of RS opening a region though tickleOpening fails due to znode version mismatch (Ramkrishna) tedyu : Files : * /hbase/trunk/CHANGES.txt * /hbase/trunk/src/main/java/org/apache/hadoop/hbase/regionserver/handler/OpenRegionHandler.java
13. Integrated in HBase-0.92 #15 (See [https://builds.apache.org/job/HBase-0.92/15/]) HBASE-4452 Possibility of RS opening a region though tickleOpening fails due to znode version mismatch (Ramkrishna) tedyu : Files : * /hbase/branches/0.92/CHANGES.txt * /hbase/branches/0.92/src/main/java/org/apache/hadoop/hbase/regionserver/handler/OpenRegionHandler.java
14. This issue was closed as part of a bulk closing operation on 2015-11-20. All issues that have been resolved and where all fixVersions have been released have been closed (following discussions on the mailing list).