

**git\_comments:**

1. only a logical error can cause here. Printing it out to make debugging easier

**git\_commits:**

1. **summary:** HBASE-3889 NPE in Distributed Log Splitting  
**message:** HBASE-3889 NPE in Distributed Log Splitting git-svn-id: <https://svn.apache.org/repos/asf/hbase/trunk@1136659> 13f79535-47bb-0310-9956-ffa450edef68

**github\_issues:****github\_issues\_comments:****github\_pulls:****github\_pulls\_comments:****github\_pulls\_reviews:****jira\_issues:**

1. **summary:** NPE in Distributed Log Splitting  
**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.
2. **summary:** NPE in Distributed Log Splitting  
**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.
3. **summary:** NPE in Distributed Log Splitting  
**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for

```
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C
org.apache.zookeeper.KeeperException$BadVersionException: KeeperErrorCode = BadVersion for
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access$100(SplitLogWorker.java:68) at
org.apache.hadoop.hbase.regionserver.SplitLogWorker$2.progress(SplitLogWorker.java:265) at
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at
org.apache.hadoop.hbase.regionserver.SplitLogWorker$1.exec(SplitLogWorker.java:113) at
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably
be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as
this never came to an end.
```

#### 4. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

**label:** code-design

#### 5. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

**label:** code-design

#### 6. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at

org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

**label:** code-design

7. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist:  
hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat}  
The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) {  
wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }  
wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN  
org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at  
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

8. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist:  
hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat}  
The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) {  
wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }  
wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN  
org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at  
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

**label:** test

9. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist:  
hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat}  
The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) {  
wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }  
wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN  
org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at  
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at

org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably  
be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as  
this never came to an end.

**label:** test

10. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for  
example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist:  
hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat}  
The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) {  
wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }  
wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What  
made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that  
was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are  
other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN  
org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at  
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably  
be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as  
this never came to an end.

11. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for  
example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist:  
hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat}  
The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) {  
wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }  
wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What  
made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that  
was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are  
other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN  
org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at  
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably  
be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as  
this never came to an end.

12. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for  
example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist:  
hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat}  
The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) {  
wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }  
wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What  
made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that  
was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are  
other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN  
org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for  
/hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C  
at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at  
org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at  
org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at  
org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably  
be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as  
this never came to an end.

13. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

14. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

15. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } } wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

**label:** code-design

16. **summary:** NPE in Distributed Log Splitting

**description:** There is an issue with the log splitting under the specific condition of edits belonging to a non existing region (which went away after a split for example). The HLogSplitter fails to check the condition, which is handled on a lower level, logging manifests it as {noformat} 2011-05-16 13:56:10,300 INFO org.apache.hadoop.hbase.regionserver.wal.HLogSplitter: This region's directory doesn't exist: hdfs://localhost:8020/hbase/userstable/30c4d0a47703214845d0676d0c7b36f0. It is very likely that it was already split so it's safe to discard those edits. {noformat} The code returns a null reference which is not check in HLogSplitter.splitLogFileToTemp(): {code} ... WriterAndPath wap = (WriterAndPath)o; if (wap == null) { wap = createWAP(region, entry, rootDir, tmpname, fs, conf); if (wap == null) { logWriters.put(region, BAD\_WRITER); } else { logWriters.put(region, wap); } }

wap.w.append(entry); ... {code} The createWAP does return "null" when the above message is logged based on the obsolete region reference in the edit. What made this difficult to detect is that the error (and others) are silently ignored in SplitLogWorker.grabTask(). I added a catch and error logging to see the NPE that was caused by the above. {code} ... break; } } catch (Exception e) { LOG.error("An error occurred.", e); } finally { if (t > 0) { ... {code} As a side note, there are other errors/asserts triggered that this try/finally not handles. For example {noformat} 2011-05-16 13:58:30,647 WARN org.apache.hadoop.hbase.regionserver.SplitLogWorker: BADVERSION failed to assert ownership for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C org.apache.zookeeper.KeeperException\$BadVersionException: KeeperErrorCode = BadVersion for /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C at org.apache.zookeeper.KeeperException.create(KeeperException.java:106) at org.apache.zookeeper.KeeperException.create(KeeperException.java:42) at org.apache.zookeeper.ZooKeeper.setData(ZooKeeper.java:1038) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.ownTask(SplitLogWorker.java:329) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.access\$100(SplitLogWorker.java:68) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$2.progress(SplitLogWorker.java:265) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:432) at org.apache.hadoop.hbase.regionserver.wal.HLogSplitter.splitLogFileToTemp(HLogSplitter.java:354) at org.apache.hadoop.hbase.regionserver.SplitLogWorker\$1.exec(SplitLogWorker.java:113) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.grabTask(SplitLogWorker.java:260) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.taskLoop(SplitLogWorker.java:191) at org.apache.hadoop.hbase.regionserver.SplitLogWorker.run(SplitLogWorker.java:164) at java.lang.Thread.run(Thread.java:680) {noformat} This should probably be handled - or at least documented - in another issue? The NPE made the log split end and the SplitLogManager add an endless amount of RESCAN entries as this never came to an end.

#### jira\_issues\_comments:

1. Trivial patch, skips null reference, discarding the obsolete edit.
2. After adding the patch I got eventually this {noformat} ... 2011-05-16 14:15:31,428 DEBUG org.apache.hadoop.hbase.monitoring.MonitoredTask: markComplete: processed 50012 edits across 47 regions threw away edits for 13 regions log file = hdfs://localhost/hbase/.logs/10.0.0.65,60020,1305406356765/10.0.0.65%2C60020%2C1305406356765.1305409968389 is corrupted = false 2011-05-16 14:15:31,428 DEBUG org.apache.hadoop.hbase.regionserver.SplitLogWorker: After Exec Status: DONE 2011-05-16 14:15:31,429 INFO org.apache.hadoop.hbase.regionserver.SplitLogWorker: successfully transitioned task /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C to final state done 2011-05-16 14:15:31,429 INFO org.apache.hadoop.hbase.regionserver.SplitLogWorker: worker 10.0.0.64,60020,1305546944326 done with task /hbase/splitlog/hdfs%3A%2F%2Flocalhost%2Fhbase%2F.logs%2F10.0.0.65%2C60020%2C1305406356765%2F10.0.0.65%252C60020%252C1305406356765.13C in 879381ms ... {noformat} This was a 15GB edits that lingered since I had a crashed YSCB bulk load. The extra messages above are from log statements I added everywhere to see what is going on (might make sense to add a debug level output of MonitoredTaskImpl.setStatus(), was helpful here to see where it got lost).
3. Unit test?
4. **body:** Thanks Lars for finding and providing a fix for this issue. I have a few minor comments on the patch ... The following isn't really needed, the earlier check you put in should be good enough. {code} + if (wap == null) { + continue; + } {code} It might be better to catch Throwable in SplitLogWorker.run() and print the Unexpected Error message there. It might not be a good thing to ignore an unexpected exception in SplitLogWorker.grabTask() and continue. {noformat} +++ src/main/java/org/apache/hadoop/hbase/regionserver/SplitLogWorker.java (working copy) @@ -297,6 +297,8 @@ } break; } + } catch (Exception e) { + LOG.error("An error occurred.", e); } finally { if (t > 0) { LOG.info("worker " + serverName + " done with task " + path + {noformat} **label:** code-design
5. **body:** Hi Prakash, the issue I had was a second RegionServer process running and still giving me this error. I only had the extra "continue", but it wasn't picked up. I forgot to remove the second "if". I agree on the error handling, I just added the log to get some info. I guess I shouldn't have made the patch available through JIRA but simply attach it for someone to pick up. I am on a tight deadline right now and won't be able to work on a unit test - although I would love to help out, once I get my current assignment completed. **label:** code-design
6. **body:** Patch is too simplistic and needs little cleanup and better error handling, plus should have a unit test. **label:** code-design
7. Marking critical. I'll pick up where Lars left off (with some help from Prakash). Thanks Lars.
8. **body:** Hi, I am working with Prakash on this. I hope to soon post a unit test. Anirudh **label:** test
9. **body:** Submitting a unit test for the above issue **label:** test
10. @Anirudh Test looks good. Do you want to include in your patch Lars fix or you want me to apply that separate? Do you want to redo it addressing issues raised by Prakash (while considering Lars' follow up?)
11. Submitting patch with Lars fix, including Prakash's comments and my unit test.
12. Thank you for the patch Anirudh (I added you as contributor and assigned you this issue). Nice test. Committed to TRUNK.
13. Thanks Stack and Prakash. I look forward to contributing more in the future.
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).