

git_comments:**git_commits:**

1. **summary:** NIFI-1684 This closes #302. fixed random partitioner initialization

message: NIFI-1684 This closes #302. fixed random partitioner initialization Signed-off-by: joewitt <joewitt@apache.org>

github_issues:**github_issues_comments:****github_pulls:**

1. **title:** NIFI-1684 fixed NPE, added tests
body:

github_pulls_comments:**github_pulls_reviews:****jira_issues:**

1. **summary:** PutKafka tuning past refactoring

description: The Kafka broker and zookeeper instance being used were replaced while NiFi was running. During that time PutKafka had NullPointerExceptions such as: {quote} 2016-03-24 05:12:58,427 WARN [Timer-Driven Process Thread-5] o.a.n.c.t.ContinuallyRunProcessorTask Administratively Yielding PutKafka[id=f8b2f669-fec5-3b26-ad2b-bca dff0c6543] due to uncaught Exception: java.lang.NullPointerException 2016-03-24 05:12:58,429 WARN [Timer-Driven Process Thread-5] o.a.n.c.t.ContinuallyRunProcessorTask java.lang.NullPointerException: null at java.lang.String.<init>(String.java:566) ~[na:1.8.0_65] at org.apache.nifi.processors.kafka.SplittableMessageContext.getKeyBytesAsString(SplittableMessageContext.java:105) ~[na:na] at org.apache.nifi.processors.kafka.PutKafka.buildFailedFlowFileAttributes(PutKafka.java:395) ~[na:na] at org.apache.nifi.processors.kafka.PutKafka.onTrigger(PutKafka.java:308) ~[na:na] at org.apache.nifi.processor.AbstractProcessor.onTrigger(AbstractProcessor.java:27) ~[nifi-api-0.6.0.jar:0.6.0] at org.apache.nifi.controller.StandardProcessorNode.onTrigger(StandardProcessorNode.java:1057) ~[nifi-framework-core-0.6.0.jar:0.6.0] at org.apache.nifi.controller.tasks.ContinuallyRunProcessorTask.call(ContinuallyRunProcessorTask.java:136) [nifi-framework-core-0.6.0.jar:0.6.0] at org.apache.nifi.controller.tasks.ContinuallyRunProcessorTask.call(ContinuallyRunProcessorTask.java:47) [nifi-framework-core-0.6.0.jar:0.6.0] at org.apache.nifi.controller.scheduling.TimerDrivenSchedulingAgent\$1.run(TimerDrivenSchedulingAgent.java:123) [nifi-framework-core-0.6.0.jar:0.6.0] at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:511) [na:1.8.0_65] at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308) [na:1.8.0_65] at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.access\$301(ScheduledThreadPoolExecutor.java:180) [na:1.8.0_65] at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294) [na:1.8.0_65] at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) [na:1.8.0_65] at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) [na:1.8.0_65] at java.lang.Thread.run(Thread.java:745) [na:1.8.0_65] {quote} But, GetKafka had no errors, stopped functioning, and became unresponsive to attempts to stop it. The 30 sec invoke quietly mechanism didn't seem to address the issue either presumably because the thread is stuck on some object monitor. I tried to stop it so I could restart it but in attempting to start it NiFi blocked me saying it was in STOPPING state. So stack dump taken and this appears relevant: {quote} "StandardProcessScheduler Thread-8" Id=142 BLOCKED on java.lang.Object@17820e22 at kafka.consumer.ZookeeperConsumerConnector.commitOffsets(ZookeeperConsumerConnector.scala:295) at kafka.javaapi.consumer.ZookeeperConsumerConnector.commitOffsets(ZookeeperConsumerConnector.scala:111) at org.apache.nifi.processors.kafka.GetKafka.shutdownConsumer(GetKafka.java:296) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:497) at org.apache.nifi.util.ReflectionUtils.invokeMethodsWithAnnotations(ReflectionUtils.java:137) at org.apache.nifi.util.ReflectionUtils.invokeMethodsWithAnnotations(ReflectionUtils.java:125) at org.apache.nifi.util.ReflectionUtils.invokeMethodsWithAnnotations(ReflectionUtils.java:233) at org.apache.nifi.util.ReflectionUtils.invokeMethodsWithAnnotations(ReflectionUtils.java:85) at org.apache.nifi.controller.StandardProcessorNode\$2.run(StandardProcessorNode.java:1330) at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:511) at java.util.concurrent.FutureTask.run(FutureTask.java:266) at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.access\$201(ScheduledThreadPoolExecutor.java:180) at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:293) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at java.lang.Thread.run(Thread.java:745)

Number of Locked Synchronizers: 1 - java.util.concurrent.ThreadPoolExecutor\$Worker@1f4d2630 {quote} Then here i see the reference to that object in another stack: {quote} "kafka-consumer-scheduler-0" Id=213 TIMED_WAITING on java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@5e93adf at sun.misc.Unsafe.park(Native Method) at java.util.concurrent.locks.LockSupport.parkUntil(LockSupport.java:256) at java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject.awaitUntil(AbstractQueuedSynchronizer.java:2120) at org.I0Itec.zkclient.ZkClient.waitForKeeperState(ZkClient.java:636) at org.I0Itec.zkclient.ZkClient.waitForConnected(ZkClient.java:619) at org.I0Itec.zkclient.ZkClient.waitForConnected(ZkClient.java:615) at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:679) at org.I0Itec.zkclient.ZkClient.writeDataReturnStat(ZkClient.java:813) at org.I0Itec.zkclient.ZkClient.writeData(ZkClient.java:808) at org.I0Itec.zkclient.ZkClient.writeData(ZkClient.java:777) at kafka.utils.ZkUtils\$.updatePersistentPath(ZkUtils.scala:326) at kafka.consumer.ZookeeperConsumerConnector.commitOffsetToZooKeeper(ZookeeperConsumerConnector.scala:283) at kafka.consumer.ZookeeperConsumerConnector\$\$anonfun\$5.apply(ZookeeperConsumerConnector.scala:304) at kafka.consumer.ZookeeperConsumerConnector\$\$anonfun\$5.apply(ZookeeperConsumerConnector.scala:303) at scala.collection.immutable.Map\$Map1.foreach(Map.scala:109) at kafka.consumer.ZookeeperConsumerConnector.commitOffsets(ZookeeperConsumerConnector.scala:303) - waiting on java.lang.Object@17820e22 at kafka.consumer.ZookeeperConsumerConnector.autoCommit(ZookeeperConsumerConnector.scala:271) at kafka.consumer.ZookeeperConsumerConnector\$\$anonfun\$1.apply\$mcV\$sp(ZookeeperConsumerConnector.scala:134) at kafka.utils.KafkaScheduler\$\$anonfun\$1.apply\$mcV\$sp(KafkaScheduler.scala:99) at kafka.utils.Utils\$\$anon\$1.run(Utils.scala:54) at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:511) at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308) at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.access\$301(ScheduledThreadPoolExecutor.java:180) at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at java.lang.Thread.run(Thread.java:745) Number of Locked Synchronizers: 1 - java.util.concurrent.ThreadPoolExecutor\$Worker@4a713b18 {quote} Which is waiting on {quote} "9e31cdb4-9685-4b04-9164-1d737edd5f31_52.90.60.74-1458794505915-6302e1bb-leader-finder-thread" Id=233 TIMED_WAITING on java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@5e93adf at sun.misc.Unsafe.park(Native Method) at java.util.concurrent.locks.LockSupport.parkUntil(LockSupport.java:256) at java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject.awaitUntil(AbstractQueuedSynchronizer.java:2120) at org.I0Itec.zkclient.ZkClient.waitForKeeperState(ZkClient.java:636) at org.I0Itec.zkclient.ZkClient.waitForConnected(ZkClient.java:619) at org.I0Itec.zkclient.ZkClient.waitForConnected(ZkClient.java:615) at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:679) at org.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:413) at org.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:409) at kafka.utils.ZkUtils\$.getChildrenParentMayNotExist(ZkUtils.scala:469) at kafka.utils.ZkUtils\$.getAllBrokersInCluster(ZkUtils.scala:81) at kafka.consumer.ConsumerFetcherManager\$LeaderFinderThread.doWork(ConsumerFetcherManager.scala:65) at kafka.utils.ShutdownableThread.run(ShutdownableThread.scala:60) Number of Locked Synchronizers: 1 - java.util.concurrent.locks.ReentrantLock\$NonfairSync@50bc97c7 {quote}

jira_issues_comments:

1. i should add that in the case of PutKafka once the broker came back up things automatically restored themselves and got moving again. While it was down and these NPEs were hitting it administratively yielded as desired. So really just a matter of better handling the case and giving the user something other than an NPE to ponder. For GetKafka though the behavior is more problematic. Given where these issues appear to live unclear what we can do for such a case.
2. that exception also occurred when using delimiter feature. So now more concerning
3. [~joewitt] Let me start from GetKafka issue as it's simple. Yes indeed we've introduced few hooks to allow Kafka processors to show signs of life in the events Kafka deadlocks. We did it for _OnSchedule and OnTrigger_. This particular stack leads to _OnStopped_ method which doesn't have those hooks. Given that StandardProcessNode invokes all OnStopped *quietly*, perhaps this could be addressed at the framework level rather than at Kafka (the same way we did OnSchedule) to basically invoke OnStopped async and wait on a Future to complete. As for PutKafka, the PR is coming
4. GitHub user olegz opened a pull request: <https://github.com/apache/nifi/pull/302> NIFI-1684 fixed NPE, added tests You can merge this pull request into a Git repository by running: \$ git pull https://github.com/olegz/nifi NIFI-1684 Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/nifi/pull/302.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #302 ---- commit 409ad1051fc44a0ce8acf1a59ba8dac5261a688e Author: Oleg Zhurakousky <oleg@suitcase.io> Date: 2016-03-24T11:37:58Z NIFI-1684 fixed NPE, added tests ----
5. One other issue {code} java.lang.NullPointerException: null at org.apache.nifi.processors.kafka.PutKafka.buildMessageContext(PutKafka.java:411) ~[na:na] at org.apache.nifi.processors.kafka.PutKafka.onTrigger(PutKafka.java:293) ~[na:na] at org.apache.nifi.processor.AbstractProcessor.onTrigger(AbstractProcessor.java:27) ~[nifi-api-0.5.1.1.2.1-37.jar:0.5.1.1.2.1-37] ... {code}
6. Also, just noticed that FF needs to be penalized on failure, fixing
7. Also, since it's still open I am fixing partitioner issue that was reported in the mailing list on 02/25/2016 {code} Caused by: java.lang.IllegalStateException: Failed to create partitioner at org.apache.nifi.processors.kafka.KafkaPublisher.<init>

(KafkaPublisher.java:75) ~[na:na] at org.apache.nifi.processors.kafka.PutKafka.createKafkaPublisher(PutKafka.java:282) ~
[na:na] ... 14 common frames omitted Caused by: java.lang.InstantiationException: kafka.producer.DefaultPartitioner at
java.lang.Class.newInstance(Class.java:427) ~[na:1.8.0_45] at org.apache.nifi.processors.kafka.KafkaPublisher.<init>
(KafkaPublisher.java:70) ~[na:na] ... 15 common frames omitted Caused by: java.lang.NoSuchMethodException:
kafka.producer.DefaultPartitioner.<init>() at java.lang.Class.getConstructor0(Class.java:3082) ~[na:1.8.0_45] at
java.lang.Class.newInstance(Class.java:412) ~[na:1.8.0_45] ... 16 common frames omitted {code}

8. [~ozhurakousky] i am comfortable with the change being included here but would ask that you please update the ticket subject and description to reflect that it is correcting defect found with Kafka processors. It is really important that these ticket subjects be valid.
9. GitHub user olegz opened a pull request: <https://github.com/apache/nifi/pull/308> NIFI-1684-B fixed ZKClient connection leak
You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/olegz/nifi> tmp Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/nifi/pull/308.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #308 ---- commit b0dfdf4c484ff9ff51bf16bab0c076c697a3d9a7 Author: Oleg Zhurakousky <oleg@suitcase.io> Date: 2016-03-29T01:48:37Z NIFI-1684-B fixed ZKClient connection leak ----
10. Seeing far better behavior with these patches applied. Probably warrants an 0.6.1 release. Need to check with community on whether it would rather go 0.7.0 to pick up already provided new features (amazon procs, split text concept) or whether we should do an 0.6.1. +1 will merge shortly and close both prs.
11. Commit e35c40b0fd6c707e4c2f6cbf1bda7c6ab610ce3c in nifi's branch refs/heads/master from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=e35c40b>] NIFI-1684 fixed NPE, added tests Signed-off-by: joewitt <joewitt@apache.org>
12. Commit 8d960f52436a516776d6fe775be6c6f47645078b in nifi's branch refs/heads/master from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=8d960f5>] NIFI-1684 fixed NPE in PutKafka when retrieving key attribute bytes Signed-off-by: joewitt <joewitt@apache.org>
13. Commit 1292581ec822f7e595e037aad437c2bb837b83eb in nifi's branch refs/heads/master from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=1292581>] NIFI-1684 added penalization on failure Signed-off-by: joewitt <joewitt@apache.org>
14. Commit 9912f18de54c6d8d7d7aaadd704199061e259330 in nifi's branch refs/heads/master from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=9912f18>] NIFI-1684 This closes #302. fixed random partitioner initialization Signed-off-by: joewitt <joewitt@apache.org>
15. Commit c3d54ab7246fba5ea1432d949c79f97f0e97f25c in nifi's branch refs/heads/master from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=c3d54ab>] NIFI-1684 This closes #308. fixed ZKClient connection leak Signed-off-by: joewitt <joewitt@apache.org>
16. Github user asfgit closed the pull request at: <https://github.com/apache/nifi/pull/308>
17. Github user asfgit closed the pull request at: <https://github.com/apache/nifi/pull/302>
18. Commit be5c95e5e93bd492c99889a3f550091fb2c4a48f in nifi's branch refs/heads/support/nifi-0.6.x from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=be5c95e>] NIFI-1684 fixed NPE, added tests Signed-off-by: joewitt <joewitt@apache.org>
19. Commit f0546d04872d202043834ff398c31420347f95b7 in nifi's branch refs/heads/support/nifi-0.6.x from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=f0546d0>] NIFI-1684 fixed NPE in PutKafka when retrieving key attribute bytes Signed-off-by: joewitt <joewitt@apache.org>
20. Commit 7561fa53e4e7ca4e3256d0de8491cc6e5b93f76e in nifi's branch refs/heads/support/nifi-0.6.x from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=7561fa5>] NIFI-1684 added penalization on failure Signed-off-by: joewitt <joewitt@apache.org>
21. Commit 89567ebfba3b6d4cf42b04aa981085b1f0e3564e in nifi's branch refs/heads/support/nifi-0.6.x from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=89567eb>] NIFI-1684 This closes #302. fixed random partitioner initialization Signed-off-by: joewitt <joewitt@apache.org>
22. Commit 552d831807efc06b714759db9ecb71cc41142826 in nifi's branch refs/heads/support/nifi-0.6.x from [~ozhurakousky] [<https://git-wip-us.apache.org/repos/asf?p=nifi.git;h=552d831>] NIFI-1684 This closes #308. fixed ZKClient connection leak Signed-off-by: joewitt <joewitt@apache.org>