

git_comments:**git_commits:**

1. **summary:** [SPARK-6014] [CORE] [HOTFIX] Add try-catch block around ShutDownHook
message: [SPARK-6014] [CORE] [HOTFIX] Add try-catch block around ShutDownHook Add a try/catch block around removeShutDownHook else IllegalStateException thrown in YARN cluster mode (see <https://github.com/apache/spark/pull/4690>) cc andrewor14, srowen Author: Nishkam Ravi <nravi@cloudera.com> Author: nishkamravi2 <nishkamravi@gmail.com> Author: nravi <nravi@c1704.halxg.cloudera.com> Closes #5672 from nishkamravi2/master_nravi and squashes the following commits: 0f1abd0 [nishkamravi2] Update Utils.scala 474e3bf [nishkamravi2] Update DiskBlockManager.scala 97c383e [nishkamravi2] Update Utils.scala 8691e0c [Nishkam Ravi] Add a try/catch block around Utils.removeShutdownHook 2be1e76 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 1c13b79 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi bad4349 [nishkamravi2] Update Main.java 36a6f87 [Nishkam Ravi] Minor changes and bug fixes b7f4ae7 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 4a45d6a [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 458af39 [Nishkam Ravi] Locate the jar using getLocation, obviates the need to pass assembly path as an argument d9658d6 [Nishkam Ravi] Changes for SPARK-6406 ccdc334 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 3faa7a4 [Nishkam Ravi] Launcher library changes (SPARK-6406) 345206a [Nishkam Ravi] spark-class merge Merge branch 'master_nravi' of <https://github.com/nishkamravi2/spark> into master_nravi ac58975 [Nishkam Ravi] spark-class changes 06bfeb0 [nishkamravi2] Update spark-class 35af990 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 32c3ab3 [nishkamravi2] Update AbstractCommandBuilder.java 4bd4489 [nishkamravi2] Update AbstractCommandBuilder.java 746f35b [Nishkam Ravi] "hadoop" string in the assembly name should not be mandatory (everywhere else in spark we mandate spark-assembly*hadoop*.jar) bfe96e0 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi ee902fa [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi d453197 [nishkamravi2] Update NewHadoopRDD.scala 6f41a1d [nishkamravi2] Update NewHadoopRDD.scala 0ce2c32 [nishkamravi2] Update HadoopRDD.scala f7e33c2 [Nishkam Ravi] Merge branch 'master_nravi' of <https://github.com/nishkamravi2/spark> into master_nravi ba1eb8b [Nishkam Ravi] Try-catch block around the two occurrences of removeShutDownHook. Deletion of semi-redundant occurrences of expensive operation inShutdown. 71d0e17 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 494d8c0 [nishkamravi2] Update DiskBlockManager.scala 3c5ddba [nishkamravi2] Update DiskBlockManager.scala f0d12de [Nishkam Ravi] Workaround for IllegalStateException caused by recent changes to BlockManager.stop 79ea8b4 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi b446edc [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 5c9a4cb [nishkamravi2] Update TaskSetManagerSuite.scala 535295a [nishkamravi2] Update TaskSetManager.scala 3e1b616 [Nishkam Ravi] Modify test for maxResultSize 9f6583e [Nishkam Ravi] Changes to maxResultSize code (improve error message and add condition to check if maxResultSize > 0) 5f8f9ed [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi 636a9ff [nishkamravi2] Update YarnAllocator.scala 8f76c8b [Nishkam Ravi] Doc change for yarn memory overhead 35daa64 [Nishkam Ravi] Slight change in the doc for yarn memory overhead 5ac2ec1 [Nishkam Ravi] Remove out dac1047 [Nishkam Ravi] Additional documentation for yarn memory overhead issue 42c2c3d [Nishkam Ravi] Additional changes for yarn memory overhead issue 362da5e [Nishkam Ravi] Additional changes for yarn memory overhead c726bd9 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi f00fa31 [Nishkam Ravi] Improving logging for AM memoryOverhead 1cf2d1e [nishkamravi2] Update YarnAllocator.scala ebcde10 [Nishkam Ravi] Modify default YARN memory_overhead-- from an additive constant to a multiplier (redone to resolve merge conflicts) 2e69f11 [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> into master_nravi efd688a [Nishkam Ravi] Merge branch 'master' of <https://github.com/apache/spark> 2b630f9 [nravi] Accept memory input as "30g", "512M" instead of an int value, to be consistent with rest of Spark 3bf8fad [nravi] Merge branch 'master' of <https://github.com/apache/spark> 5423a03 [nravi] Merge branch 'master' of <https://github.com/apache/spark> eb663ca [nravi] Merge branch 'master' of <https://github.com/apache/spark> df2aeb1 [nravi] Improved fix for ConcurrentModificationIssue (Spark-

1097, Hadoop-10456) 6b840f0 [nravi] Undo the fix for SPARK-1758 (the problem is fixed) 5108700 [nravi] Fix in Spark for the Concurrent thread modification issue (SPARK-1097, HADOOP-10456) 681b36f [nravi] Fix for SPARK-1758: failing test org.apache.spark.JavaAPISuite.wholeTextFiles

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-6014] [core][hotfix] Add try-catch block around ShutdownHook
body: Add a try/catch block around removeShutdownHook else IllegalStateException thrown in YARN cluster mode (see <https://github.com/apache/spark/pull/4690>) cc @andrewor14, @srowen

github_pulls_comments:

1. LGTM, be perhaps check for `Utils.inShutdown` instead? Also, could you change the PR title to reflect the actual change?
2. See Utils.inShutdown discussion in 4690.
3. Can this be tested please. Thanks.
4. > See Utils.inShutdown discussion in 4690. The part about it being expensive? This is hardly a performance sensitive path. It could also be made faster by first checking `shutdownHooks.shutdown` before trying the `Runtime.blah` route. But in any case, not a big deal.
5. I would be fine with 'not explicitly throwing an exception' and keeping the try-catch block.
6. LGTM. `try..catch` seems unnecessary now but doesn't hurt either.
7. Why keep the block? what exception is expected now and why is it safe to squash it (without logging even)?
8. A case of erring on the conservative side: the implementation of Utils.removeShutdownHook could change over time. The invocation from DiskBlockManager should not have to rely on it being exception-free.
9. With regards to logging, I thought you had recommended not logging (in 4690) . Same reasons apply?
10. There I was talking about a normal, expected exception. If this method changes to not throw an exception that we think of as normal and expect-able, then whatever this catches is surprising and should be logged. By this logic we should catch a lot of exceptions in a lot of places... this makes me believe that the author expected a particular exception and specially knows it can be silently ignored. That doesn't seem to be the argument anymore.
11. The original PR in 4690 didn't make assumptions about the type of Exception thrown and logged it, I guess we were thinking slightly differently back then too. Agreed: generalizing and logging seems like the best way forward.
12. In general I'd argue for not just catching exceptions broadly as a defense. This is a `stop()` method though, and often the right behavior is to just keep going even if stopping failed due to some I/O problem or messed up state. Here I don't see what exception this method can throw. I'd still find `try-catch` surprising and would not expect it.
13. Since we're talking exceptions, `SparkShutdownHookManager.runAll` could be modified to catch exceptions for individual hooks and ignore them. It already logs them, but `logUncaughtExceptions` re-throws, so an exception would cause downstream hooks to not be executed.
14. @srowen I think I'd approach this one as why-not (especially with logging enabled). This particular call site has warranted two separate PRs attempting to add a try-catch block because the code in Utils changed and thus stands out from the rest of Spark code in that regard?
15. @vanzin Are we suggesting adding a try-catch block inside the while loop in runAll or deleting throw's in logUncaughtExceptions ?
16. @nishkamravi2 just adding `Try()` around the `logUncaughtExceptions` call should be sufficient.
17. While we are at it: any particular reason for using PriorityQueue or can we replace it by ConcurrentLinkedQueue or something and get rid of a few synchronized's ?
18. Hooks have a priority so that they can choose the order they're executed in. e.g. if some hook really needs to execute before another one, that's possible. Similar to how the Spark hook needs to execute before the HDFS one to avoid exceptions. The locking shouldn't cause issues - this is not a hot path where avoiding locks brings any benefit. Outside of shutdown, there's just a handful of places where the methods are

called. During shutdown, everything happens from `runAll`, so any re-entrant calls already have the lock held and thus can go through.

19. @vanzin Sounds reasonable.
20. @nishkamravi2 this isn't the same method or exception handling as before. The question is what's right to do given the current state of the code. It feels a little arbitrary to wrap this invocation but not others, but it's a `stop()` method and at least logs it, so won't argue it further.
21. ok to test
22. [Test build #701 has started]
(<https://amplab.cs.berkeley.edu/jenkins/job/NewSparkPullRequestBuilder/701/consoleFull>) for PR 5672 at commit [`0f1abd0`]
(<https://github.com/apache/spark/commit/0f1abd01287cd33aa73a0b5574f95369b8d42910>).
23. This test actually succeeded normally, just wasn't able to post back here. Going ahead.

github_pulls_reviews:

1. Can this simply not throw an exception?
2. What would it do instead? Just ignore the call and return `false`? Probably not a big deal in practice, but it does overload the return value to mean "either the hook doesn't exist, or I can't remove it at this time".
3. Yes. Asking to remove a hook always races with shutdown, so I suppose code can never guarantee it's been able to remove a hook before shutdown. It can't be sure it has successfully removed the hook before it executes. So it doesn't seem too bad to just try to remove the hook, which might or might not have executed and discarded already, and return normally either way. That is do you want to prohibit removal of a hook that hasn't run, even during shutdown? meh, might be a simpler way to resolve this.
4. Hmm. It should be ok to allow removing the hooks during shutdown.
`SparkShutdownHookManager.runAll` is safe against modifications to the underlying queue.

jira_issues:

1. **summary:** Bug in Configuration.java exposed by Spark (ConcurrentModificationException)
description: The following exception occurs non-deterministically:
java.util.ConcurrentModificationException at
java.util.HashMap\$HashIterator.nextEntry(HashMap.java:926) at
java.util.HashMap\$KeyIterator.next(HashMap.java:960) at
java.util.AbstractCollection.addAll(AbstractCollection.java:341) at java.util.HashSet.<init>
(HashSet.java:117) at org.apache.hadoop.conf.Configuration.<init>(Configuration.java:671) at
org.apache.hadoop.mapred.JobConf.<init>(JobConf.java:439) at
org.apache.spark.rdd.HadoopRDD.getJobConf(HadoopRDD.scala:110) at
org.apache.spark.rdd.HadoopRDD\$\$anon\$1.<init>(HadoopRDD.scala:154) at
org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:149) at
org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:64) at
org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241) at
org.apache.spark.rdd.RDD.iterator(RDD.scala:232) at
org.apache.spark.rdd.MappedRDD.compute(MappedRDD.scala:31) at
org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241) at
org.apache.spark.rdd.RDD.iterator(RDD.scala:232) at
org.apache.spark.rdd.MappedRDD.compute(MappedRDD.scala:31) at
org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241) at
org.apache.spark.rdd.RDD.iterator(RDD.scala:232) at
org.apache.spark.rdd.MappedRDD.compute(MappedRDD.scala:31) at
org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241) at
org.apache.spark.rdd.RDD.iterator(RDD.scala:232) at
org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:34) at
org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:241) at
org.apache.spark.rdd.RDD.iterator(RDD.scala:232) at
org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:161) at
org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:102) at
org.apache.spark.scheduler.Task.run(Task.scala:53) at
org.apache.spark.executor.Executor\$TaskRunner\$\$anonfun\$run\$1.apply\$mcV\$sp(Executor.scala:213) at
org.apache.spark.deploy.SparkHadoopUtil\$\$anon\$1.run(SparkHadoopUtil.scala:42) at

org.apache.spark.deploy.SparkHadoopUtil\$.anon\$1.run(SparkHadoopUtil.scala:41) at
java.security.AccessController.doPrivileged(Native Method) at
javax.security.auth.Subject.doAs(Subject.java:415) at
org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1548) at
org.apache.spark.deploy.SparkHadoopUtil.runAsUser(SparkHadoopUtil.scala:41) at
org.apache.spark.executor.Executor\$TaskRunner.run(Executor.scala:178) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145) at
java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:615) at
java.lang.Thread.run(Thread.java:744)

jira_issues_comments:

1. Patch attached.
2. Fix is to move the HashSet initialization to the synchronized block right above it. This patch can potentially be applied to all Hadoop branches.
3. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12638356/HADOOP-10456_nravi.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-common-project/hadoop-common. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3738/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3738/console> This message is automatically generated.
4. This is a fix for a non-deterministic bug (a thread synchronization issue). A unit test case for this fix would be flaky. The verification was done by (i) running mvn compile/install/test, (ii) running Spark benchmarks and making sure there are no performance regressions
5. Thank you for taking this JIRA, [~nravi]. +1(non-binding), reviewed and confirmed it's a only critical section about finalParameters without lock within Configuration instance. [~cnauroth], can you take a look, please?
6. +1 for the patch. Thank you, Nishkam and Tsuyoshi. I can't commit this right now. I'll aim to commit tomorrow. If I don't commit it in the next few days, please ping me again in case I forget. :-)
7. Thanks Tsuyoshi and Chris.
8. SUCCESS: Integrated in Hadoop-trunk-Commit #5456 (See [<https://builds.apache.org/job/Hadoop-trunk-Commit/5456/>]) HADOOP-10456. Bug in Configuration.java exposed by Spark (ConcurrentModificationException). Contributed by Nishkam Ravi. (cnauroth: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1584575>) *
/hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt *
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/conf/Configuration.java
9. I committed this to trunk, branch-2 and branch-2.4. Nishkam, thank you for contributing this bug fix. Tsuyoshi, thank you for the code review.
10. SUCCESS: Integrated in Hadoop-Yarn-trunk #529 (See [<https://builds.apache.org/job/Hadoop-Yarn-trunk/529/>]) HADOOP-10456. Bug in Configuration.java exposed by Spark (ConcurrentModificationException). Contributed by Nishkam Ravi. (cnauroth: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1584575>) *
/hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt *
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/conf/Configuration.java
11. FAILURE: Integrated in Hadoop-Mapreduce-trunk #1747 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1747/>]) HADOOP-10456. Bug in Configuration.java exposed by Spark (ConcurrentModificationException). Contributed by Nishkam Ravi. (cnauroth: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1584575>) *

/hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt *
/hadoop/common/trunk/hadoop-common-project/hadoop-
common/src/main/java/org/apache/hadoop/conf/Configuration.java
12. FAILURE: Integrated in Hadoop-Hdfs-trunk #1721 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/1721/>]) HADOOP-10456. Bug in Configuration.java exposed by Spark
(ConcurrentModificationException). Contributed by Nishkam Ravi. (cnauroth:
<http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1584575>) *
/hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt *
/hadoop/common/trunk/hadoop-common-project/hadoop-
common/src/main/java/org/apache/hadoop/conf/Configuration.java