Hadoop Map/Reduce / MAPREDUCE-6156

Fetcher - connect() doesn't handle connection refused correctly

Details

Type:	Bug	Status:	CLOSED
Priority:	Blocker	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	2.6.0
Component/s:	None		
Labels:	None		
Target Version/s:	2.6.0		
Hadoop Flags:	Reviewed		

Description

The connect() function in the fetcher assumes that whenever an IOException is thrown, the amount of time passed equals "connectionTimeout" (see code snippet below). This is incorrect. For example, in case the NM is down, an ConnectException is thrown immediately - and the catch block assumes a minute has passed when it is not the case.

```
if (connectionTimeout < 0) {
    throw new IOException("Invalid timeout "
                          + "[timeout = " + connectionTimeout + " ms]");
  } else if (connectionTimeout > 0) {
    unit = Math.min(UNIT_CONNECT_TIMEOUT, connectionTimeout);
  }
  // set the connect timeout to the unit-connect-timeout
  connection.setConnectTimeout(unit);
  while (true) {
    try {
      connection.connect();
      break;
    } catch (IOException ioe) {
      // update the total remaining connect-timeout
      connectionTimeout -= unit;

      // throw an exception if we have waited for timeout amount of time
      // note that the updated value if timeout is used here
      if (connectionTimeout == 0) {
        throw ioe;
      }

      // reset the connect timeout for the last try
```

Attachments

MAPREDUCE-6156.patch	2 kB	12/Nov/14 06:33
MAPREDUCE-6156-v2.patch	3 kB	13/Nov/14 00:27
MAPREDUCE-6156-v3.patch	3 kB	13/Nov/14 01:48

Issue Links

is duplicated by

MAPREDUCE-6157 Connect failed in shuffle (due to NM down) could break current retry logic to tolerant N... RESOLVED

relates to

MAPREDUCE-5891 Improved shuffle error handling across NM restarts CLOSED

Activity

Junping Du added a comment - 12/Nov/14 06:29

Thanks sidharta-s for reporting this. Looks like I just filed a duplicated one, and I will move the patch to here.

Hadoop QA added a comment - 12/Nov/14 07:13

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12681011/MAPREDUCE-6156.patch
against trunk revision 53f64ee.

+1 @author. The patch does not contain any @author tags.

-1 tests included. 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.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

+1 core tests. The patch passed unit tests in hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core.

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/5011//testReport/
Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/5011//console

This message is automatically generated.

Jason Darrell Lowe added a comment - 12/Nov/14 15:54

Thanks for the patch, Junping! Sorry for missing this in the review of ~~MAPREDUCE-5891~~.

Rather than throwing the original IOException when we stop retrying, it might be more informative to wrap it with a message stating how many attempts and how long we waited total to try to connect. Otherwise we could throw a socket timeout exception indicated we only waited 1ms depending upon how the attempts lined up with the retries.

Speaking of trying up to the retry timeout, we're computing a potentially lower connection timeout value based on the amount of time left in total retry time but then we could end up sleeping most of that time away just below. Arguably we should be computing the socket timeout time just before we try to connect rather than just after we received the error.

I don't think we should always ignore InterruptedException. That's probably what we'll get when the Fetcher threads are trying to be shut down, so I think we should return if `stopped==true` when that's received.

Siddharth Seth added a comment - 12/Nov/14 23:43

One other bit. Do we want to retry the connections in a tight loop if there's a ConnectException. The sleep time is computed as (currentTime-lastTime) - which is likely to be a tight loop while the NM is down. Alternates would be to sleep until the difference with 'unit' or for a shorter period - but at least some wait.

Sidharta Seethana added a comment - 13/Nov/14 00:25

Agree with what sseth said. We shouldn't use a tight loop in case of connection failures. It might also be simpler to use something like this : https://hadoop.apache.org/docs/r1.1.1/api/org/apache/hadoop/io/retry/package-summary.html instead of rolling our own retry implementation.

Junping Du added a comment - 13/Nov/14 00:27

Thanks for review and comment, jlowe and sseth!

> it might be more informative to wrap it with a message stating how many attempts and how long we waited total to try to connect.

Nice catch, address it in v2 patch.

> Arguably we should be computing the socket timeout time just before we try to connect rather than just after we received the error.

I think it could be the same in math, even for the last try in unit time, as we will sleep for the rest of unit time (like Sid suggested below) and give connection another chance. May be no much different here?

> I think we should return if stopped==true when that's received.

Sounds good. Addressed in v2 patch.

> he sleep time is computed as (currentTime-lastTime) - which is likely to be a tight loop while the NM is down.

Thanks for identifying this, I think we should sleep the rest of unit time (unit - (currentTime-lastTime)). Addressed in v2 patch.

Hadoop QA added a comment - 13/Nov/14 01:02

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12681186/MAPREDUCE-6156-v2.patch
against trunk revision d7150a1.

+1 @author. The patch does not contain any @author tags.

-1 tests included. 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.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

+1 core tests. The patch passed unit tests in hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core.

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/5015//testReport/
Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/5015//console

This message is automatically generated.

Siddharth Seth added a comment - 13/Nov/14 01:08

Looks good to me. Minor stuff: the log message needs a space "seconds" -> " seconds"
Also, the branch-2 patch will require an additional import.

Junping Du added a comment - 13/Nov/14 01:09

> We shouldn't use a tight loop in case of connection failures.

No tight loop in v2 patch now. This is easy to fix. About reuse retry policy in hadoop-common, we already have tons of retry logic that doesn't conform that (make things more complicated sometimes). Let's don't do this at this moment.

Junping Du added a comment - 13/Nov/14 01:48

Thanks sseth for review! v3 patch addressed your comments.

> Minor stuff: the log message needs a space "seconds" -> " seconds"

Nice catch! Fix it and several other places we don't have space.

> Also, the branch-2 patch will require an additional import.

I tried to apply the v3 in branch-2 and build successfully. Am I missing anything here?

Hadoop QA added a comment - 13/Nov/14 02:22

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12681205/MAPREDUCE-6156-v3.patch
against trunk revision b0a9cd3.

+1 @author. The patch does not contain any @author tags.

-1 tests included. 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.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

+1 core tests. The patch passed unit tests in hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core.

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/5016//testReport/
Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/5016//console

This message is automatically generated.

Siddharth Seth added a comment - 13/Nov/14 03:21

> I tried to apply the v3 in branch-2 and build successfully. Am I missing anything here?

My bad, was working with an older revision.

Jason Darrell Lowe added a comment - 13/Nov/14 15:38

+1 lgtm. Committing this.

Hudson added a comment - 13/Nov/14 15:51

FAILURE: Integrated in Hadoop-trunk-Commit #6533 (See https://builds.apache.org/job/Hadoop-trunk-Commit/6533/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java
- hadoop-mapreduce-project/CHANGES.txt

Jason Darrell Lowe added a comment - 13/Nov/14 16:02

Thanks to Junping for the contribution and to Sidd for additional review! I committed this to trunk, branch-2, and branch-2.6.

acmurthy this is another item that should be in the next 2.6.0 rc.

Arun Murthy added a comment - 13/Nov/14 18:50

jlowe Thanks. I've merged this into branch-2.6.0 also for hadoop-2.6.0-rc1.

Hudson added a comment - 14/Nov/14 11:32

FAILURE: Integrated in Hadoop-Yarn-trunk-Java8 #5 (See https://builds.apache.org/job/Hadoop-Yarn-trunk-Java8/5/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/CHANGES.txt
- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java

Hudson added a comment - 14/Nov/14 11:51

SUCCESS: Integrated in Hadoop-Yarn-trunk #743 (See https://builds.apache.org/job/Hadoop-Yarn-trunk/743/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java
- hadoop-mapreduce-project/CHANGES.txt

Hudson added a comment - 14/Nov/14 14:18

SUCCESS: Integrated in Hadoop-Hdfs-trunk #1933 (See https://builds.apache.org/job/Hadoop-Hdfs-trunk/1933/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java
- hadoop-mapreduce-project/CHANGES.txt

Hudson added a comment - 14/Nov/14 14:19

SUCCESS: Integrated in Hadoop-Hdfs-trunk-Java8 #5 (See https://builds.apache.org/job/Hadoop-Hdfs-trunk-Java8/5/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java
- hadoop-mapreduce-project/CHANGES.txt

Hudson added a comment - 14/Nov/14 15:16

FAILURE: Integrated in Hadoop-Mapreduce-trunk #1957 (See https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1957/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/CHANGES.txt
- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java

Hudson added a comment - 14/Nov/14 15:25

SUCCESS: Integrated in Hadoop-Mapreduce-trunk-Java8 #5 (See https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Java8/5/)
~~MAPREDUCE-6156~~. Fetcher - connect() doesn't handle connection refused correctly. Contributed by Junping Du (jlowe: rev 177e8090f5809beb3ebcb656cd0affbb3f487de8)

- hadoop-mapreduce-project/CHANGES.txt
- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/task/reduce/Fetcher.java

People

Assignee:

Junping Du

Reporter:

Sidharta Seethana

Votes:

0 Vote for this issue

Watchers:

11 Start watching this issue

Dates

Created:

12/Nov/14 01:41

Updated:

01/Dec/14 03:07

Resolved:

13/Nov/14 16:02