

Item 138

**git\_comments:**

1. happy cases
2. sad cases
3. \* \* Test parsing various types of Java heap options.

**git\_commits:**

1. **summary:** MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha)  
**message:** MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha)

**github\_issues:**

**github\_issues\_comments:**

**github\_pulls:**

**github\_pulls\_comments:**

**github\_pulls\_reviews:**

**jira\_issues:**

1. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
2. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.  
**label:** code-design
3. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.  
**label:** code-design
4. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
5. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
6. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
7. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
8. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
9. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
10. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
11. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
12. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes  
**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.
13. **summary:** JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes

**description:** It currently tries to parse the value as an integer, which blows up whenever the value is greater than 2GB and expressed in bytes.

#### jira\_issues\_comments:

1. I ran into this a few weeks ago as well. My notes on the issue: - When either of the properties: `mapreduce.map.java.opts.max.heap` `mapreduce.reduce.java.opts.max.heap` is set to a value greater than `Integer.MAX_VALUE` (or less than `Integer.MIN_VALUE`, technically), the AM log throws the exception during `TaskAttemptImpl` instantiation: 2015-04-27 16:24:39,938 FATAL [AsyncDispatcher event handler] `org.apache.hadoop.yarn.event.AsyncDispatcher: Error in dispatcher thread java.lang.NumberFormatException: For input string: "3221225472" at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65) at java.lang.Integer.parseInt(Integer.java:495) at java.lang.Integer.parseInt(Integer.java:527) at org.apache.hadoop.mapred.JobConf.parseMaximumHeapSizeMB(JobConf.java:2105) at org.apache.hadoop.mapred.JobConf.getMemoryRequired(JobConf.java:2151) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskAttemptImpl.getMemoryRequired(TaskAttemptImpl.java:563) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskAttemptImpl.<init>(TaskAttemptImpl.java:542) at org.apache.hadoop.mapred.MapTaskAttemptImpl.<init>(MapTaskAttemptImpl.java:47) at org.apache.hadoop.mapreduce.v2.app.job.impl.MapTaskImpl.createAttempt(MapTaskImpl.java:62) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl.addAttempt(TaskImpl.java:610) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl.addAndScheduleAttempt(TaskImpl.java:597) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl.access$1300(TaskImpl.java:101) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl$InitialScheduleTransition.transition(TaskImpl.java:887) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl$InitialScheduleTransition.transition(TaskImpl.java:882) at org.apache.hadoop.yarn.state.StateMachineFactory$SingleInternalArc.doTransition(StateMachineFactory.java:362) at org.apache.hadoop.yarn.state.StateMachineFactory.doTransition(StateMachineFactory.java:302) at org.apache.hadoop.yarn.state.StateMachineFactory.access$300(StateMachineFactory.java:46) at org.apache.hadoop.yarn.state.StateMachineFactory$InternalStateMachine.doTransition(StateMachineFactory.java:448) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl.handle(TaskImpl.java:648) at org.apache.hadoop.mapreduce.v2.app.job.impl.TaskImpl.handle(TaskImpl.java:100) at org.apache.hadoop.mapreduce.v2.app.MRAppMaster$TaskEventDispatcher.handle(MRAppMaster.java:1299) at org.apache.hadoop.mapreduce.v2.app.MRAppMaster$TaskEventDispatcher.handle(MRAppMaster.java:1293) at org.apache.hadoop.yarn.event.AsyncDispatcher.dispatch(AsyncDispatcher.java:173) at org.apache.hadoop.yarn.event.AsyncDispatcher$1.run(AsyncDispatcher.java:106) at java.lang.Thread.run(Thread.java:745)` Note that the exception occurs during the constructor of `TaskAttemptImpl`. - Modifying methods and members to change int to a long isn't really viable unless trunk is ready to change the `ResourceProto` definition for memory from int32 to int64 in `yarn_protos.proto`. This would be an incompatible change. I'm not sure how strongly people feel about changing proto definitions. - Capping the value might be reasonable by catching the `NumberFormatException`. Setting the value to some "uninitialized" value (like -1) had some side effects that needed to be checked for elsewhere in the code. - It's fairly easy to pass such a value to `JobConf` in the unit test and get the exception. Does anyone suggestions for how best to fix this?
2. **body:** Instead of allowing larger than 2 PiB heap (which is unrealistic anyway), a less intrusive fix could be using a long temp variable when parsing, and cast the final value back to int.  
**label:** code-design
3. **body:** Just ran into this on our development environment using CDH5.4.0: 2015-04-28 00:08:52,427 FATAL [AsyncDispatcher event handler] `org.apache.hadoop.yarn.event.AsyncDispatcher: Error in dispatcher thread java.lang.NumberFormatException: For input string: "2147483648" at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65) at java.lang.Integer.parseInt(Integer.java:499) at java.lang.Integer.parseInt(Integer.java:527) at org.apache.hadoop.mapred.JobConf.parseMaximumHeapSizeMB(JobConf.java:2105` Changing it to 2147483647 works.  
**label:** code-design
4. I hit the issue on CDH 5.4.0 too. I've opened a ticket there since this is not supposed to go in Hadoop 2.6.0. <https://issues.cloudera.org/browse/DISTRO-713>
5. Hi Casting to integer may not be a good idea, because the user might expect the same heap as he configured but actually not. It will be good if the variable is changed to long at the earliest at least in trunk. Since java support configuring the mx in bytes, it will be good if mapreduce framework can support the big values
6. bq. Casting to integer may not be a good idea, because the user might expect the same heap as he configured but actually not. The heap size in `javaopts` as parsed by this method is used ONLY to determine what the container size should be, if it is not specified. The `javaopts` string is used to spawn the tasks as is. The approach in the current patch seems reasonable to me. Comments on the patch: # Instead of using the Guava call, let us just use the simple Java type cast. # We should add unit tests for this method for various reasonable heap sizes specified in any of GB, MB, KB, B.
7. [~jasonxh] - we would like to get this issue resolved as soon as we can. Are you able to work on the follow-up patch to address the review comments. Otherwise, I ll be happy to help.

8. Agree with #2. I was easily able to use examples in TestJobConf to set a config and call the various memory methods in order to force a NumberFormatException. It should be easy to add in any combination of config settings corner cases to set.
9. Thanks [~kasha], I've updated the patch.
10. +1
11. Thanks [~jasonxh] for reporting and working on this. I just committed this to trunk.
12. FAILURE: Integrated in Hadoop-trunk-Commit #7696 (See [https://builds.apache.org/job/Hadoop-trunk-Commit/7696/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java \* hadoop-mapreduce-project/CHANGES.txt \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java
13. FAILURE: Integrated in Hadoop-Hdfs-trunk #2110 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk/2110/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/CHANGES.txt \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java
14. FAILURE: Integrated in Hadoop-Hdfs-trunk-Java8 #169 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk-Java8/169/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java \* hadoop-mapreduce-project/CHANGES.txt \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java
15. FAILURE: Integrated in Hadoop-Yarn-trunk-Java8 #178 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk-Java8/178/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java \* hadoop-mapreduce-project/CHANGES.txt
16. FAILURE: Integrated in Hadoop-Yarn-trunk #912 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk/912/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java \* hadoop-mapreduce-project/CHANGES.txt \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java
17. FAILURE: Integrated in Hadoop-Mapreduce-trunk-Java8 #179 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Java8/179/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java \* hadoop-mapreduce-project/CHANGES.txt \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java
18. FAILURE: Integrated in Hadoop-Mapreduce-trunk #2128 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk/2128/]) MAPREDUCE-6343. JobConf.parseMaximumHeapSizeMB() fails to parse value greater than 2GB expressed in bytes. (Hao Xia via kasha) (kasha: rev 519092322dd1bf71984bef1393d8e082643408cd) \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapred/JobConf.java \* hadoop-mapreduce-project/CHANGES.txt \* hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/test/java/org/apache/hadoop/mapred/TestJobConf.java