Item 159
git_comments:

1. * * Retains a strong reference to each string instance it has interned.
2. * * Interns and returns a reference to the representative instance * for any of a collection of string instances that are equal to each other. * Retains weak reference to the instance, * and so does not prevent it from being garbage-collected. * * @param sample string instance to be interned * @return weak reference to interned string instance
3. **comment:** * * Provides equivalent behavior to String.intern() to optimize performance, * whereby does not consume memory in the permanent generation.
   label: code-design
4. * * Retains a weak reference to each string instance it has interned.
5. * * Interns and returns a reference to the representative instance * for any of a collection of string instances that are equal to each other. * Retains strong reference to the instance, * thus preventing it from being garbage-collected. * * @param sample string instance to be interned * @return strong reference to interned string instance
6. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
7. * * Test the same strong reference is returned for any * of string instances that are equal to each other.
8. * * * Tests string interning {@link StringInterner}
9. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
10. * * Test the same weak reference is returned for any * of string instances that are equal to each other.
11. * * Test different references are returned for any of string * instances that are equal to each other but not interned.

git_commits:

1. **summary:** MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn)
   message: MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) git-svn-id: https://svn.apache.org/repos/asf/hadoop/common/trunk@1401473 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Counter names' memory usage can be decreased by interning

description: In our experience, most of the memory in production JTs goes to storing counter names (String objects and character arrays). Since most counter names are reused again and again, it would be a big memory savings to keep a hash set of already-used counter names within a job, and refer to the same object from all tasks.

jira_issues_comments:

1. This could be good on large systems, but as the strings go into the PermGen space, people are going to have to be playing with the JVM options to make that large enough, so it's not without some consequences.
2. **body:** Hi Steve. I am suggesting using a GettableSet<String> per JobInProgress, rather than actually calling String.intern. That way the stuff might make it to old-gen, but won't ever go to perm.
   label: code-design
3. Sounds similar in approach to HDFS-1110, which was done for filenames on DFS.
4. **body:** If it doesn't go near PermGen then it would be useful & not another source of pain, which makes it more appealing to me
   label: code-design
5. What is current status of this issue? If nobody else is already working on it, I would be very willing to start.
6. Hey Miomir. I dont think anyone's working on it. Go for it! I'd suggest looking at Guava's [Interner|http://docs.guava-libraries.googlecode.com/git/javadoc/com/google/common/collect/Interners.html#newWeakInterner()] implementation
7. Hi Todd, Thanks for suggesting Guava's Interner With the Interner, we can canonicalize counter names without filling up PermGen store, thus we don't need to use GSet<String>. Judging from the issue description, there are a few counters where string instances are created every time counter name or display name is queried. But so far, I managed to identify only following one: @InterfaceAudience.Private public static class FSCounter extends AbstractCounter { final String scheme; final FileSystemCounter key; private long value; public FSCounter(String scheme, FileSystemCounter ref) { this.scheme = scheme; key = ref; } @Override public String getName() { return NAME_JOINER.join(scheme, key.name()); } @Override public String getDisplayName() { return DISP_JOINER.join(scheme, localizeCounterName(key.name())); } Could perhaps point me to some of the remaining ones?
8. +1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12546184/MAPREDUCE-4229.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified test files. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 javadoc. The javadoc tool did not generate any warning messages. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) 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-common-project/hadoop-common 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/2869//testReport/ Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2869//console This message is automatically generated.
9. Patch involves memory saving achieved with the use of Guava's Interner implementation, including tests to verify only one copy of each distinct counter name string is stored. Please review.
10. **body:** I have a couple of questions/suggestions for the patch. Why are we using a strong interner and not a weak interner? By using the strong version I think we will have a memory leak in the history server. When someone declares a custom counter name it will never go away, even after that job's jhist file has been deleted out of HDFS and the counters are no longer accessible. I think this is even worse for some of the strings that we are interning that contain the value of the counter, not just the name of the counter. They will probably be different every time the function is called, causing some potentially very large memory leaks. But I am not really all that sure they are called from a long running process. It seems kind of odd where the interning calls are happening. I doubt it is going to save any memory at all. For example we save the name of a counter inside a counter class instance and then only intern it when we return the name from the getter method. If the intern call actually did anything it would not allow for the original name to be garbage collected, because it is still pointed to by the counter instance. The only time we really need to intern a string is when that string is the result of reading data from a stream. So this would

be for all RPC calls and anything that parses out a string from a file. In most other cases, like with strings that come directly from an ENUM or quoted string in the code they will already be interned by the runtime environment and adding this extra layer will only slow things down and actually use more memory. I think it would be preferable to start out just interning the names of the counters and counter groups as they are read from a stream, like in the case of parsing the job history files. Once that happens we can go back and evaluate if there are other places, like through RPC, that are using a lot of memory. I would hold off on the RPC, because I am not really sure how clean it is to insert this into the protocol buffer bridge code that we use. I think PB plays games with lazy parsing of the data and if we are not careful it could slow things down, or cause more memory usage.
label: code-design

11. Also the patch is against trunk not 1.X. It would be good to update the affects version and target version fields appropriately.
12. Canceling patch until my concerns are addressed, and updating target/affects fields appropriately. I targeted this for 0.23.5 because I would love to see this on branch-0.23.
13. Hi Robert, thanks for the feedback. I will implement another patch (against branch-0.23) and try to address the issues you brought up.
14. {color:green}+1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12549396/MAPREDUCE-4229-branch-0.23.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 1 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any 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 hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2932//testReport/ Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2932//console This message is automatically generated.
15. **body:** Sorry it has taken me so long to respond. I have been a bit swamped lately. The new patch looks really good. It is simple and looks like it could help a lot with the memory usage. Do you have any actual heap comparisons that you can show us? I just have a difficult time checking in a "performance" fix without some test, manual or otherwise to show the impact it is having and if there is still more that could be done in a follow up JIRA. I know that YourKit profiler has some nice Heap Dump analysis to look for duplicate strings. If you have some numbers ready that would be great otherwise I will try and find some time this week to see if I can come up with anything.
label: code-design
16. Sorry, this has been a hectic week for me too. Unfortunately, I don't have any figures to share yet but will try to capture some heap dumps over the weekend.
17. Patch that reduces memory consumption on History Server.
18. **body:** I rand some benchmarks looking at the Job History server using a jhist file for a job that had 9416 maps and 500 reducers. I then used a combination of YourKit and jhat to look at the heap savings. For Jhat I did the OQL {noformat}select sum(map(heap.objects("java.lang.String"),"sizeof(it)")){noformat} to get the size of all of the strings currently reachable on the heap. I saw that nothing changed in between the base and the first patch. Both of them had 22MB of strings in the heap. Looking at the code that was changed to do interning, the only code that uses it was rumen. It is still a good change, but it did not have the impact I was looking for. So I implemented the patch I just attached which adds in interning of Strings that are parsed out of the jhist file. This reduced the 22MB of strings to 3MB of strings. I want to do something similar for the AM, but it is more difficult to look at, and I don't think I will have time in the near future. So if someone else could review this we can check it in and file a follow up JIRA for looking at the AM.
label: code-design
19. {color:green}+1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12549767/MR-4229.txt against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 1 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1

javadoc{color}. The javadoc tool did not generate any 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 hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2941//testReport/ Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2941//console This message is automatically generated.

20. **body:** I just thought of one more thing we should do. We should make the StringInterner as @Public and @Stable. The API is simple enough I don't see much of a problem locking it down.
    label: code-design
21. {color:green}+1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12550156/MAPREDUCE-4229-branch-0.23.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 1 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any 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 hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2951//testReport/ Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2951//console This message is automatically generated.
22. Patch to annotate StringInterner with @Public and @Stable
23. The patch looks good to me and I am +1 for it, but because I wrote some of it myself it would be nice if someone else could take a look at it too.
24. **body:** I'd suggest maybe using a hard intern for some of the well-defined {{TaskAttemptImpl}} fields, but that could be a future jira. Only nit is "substirng" is misspelled in many places. Otherwise, +1.
    label: code-design
25. substring fixed
26. {color:green}+1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12550502/mr-4229.txt against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 1 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any 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 hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2962//testReport/ Console output: https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2962//console This message is automatically generated.
27. +1 Will commit shortly.
28. Integrated in Hadoop-trunk-Commit #2916 (See [https://builds.apache.org/job/Hadoop-trunk-Commit/2916/]) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401473) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401467) Result = SUCCESS daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401473 Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/StringInterner.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestStringInterner.java daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401467 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-

core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/EventReader.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/JobHistoryParser.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/util/CountersStrings.java

29. **body:** Daryn, could you update the CHANGES.txt to include Miomir Boljanovic too. He did much of the core work for this patch and deserves a lot of the credit for it.
    label: documentation
30. I've committed to trunk, branch-2, and branch-23. Thanks Miomir and Bobby!
31. Integrated in Hadoop-trunk-Commit #2918 (See [https://builds.apache.org/job/Hadoop-trunk-Commit/2918/]) Updating credits for MAPREDUCE-4229. (Revision 1401493) Result = SUCCESS daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401493 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt
32. **body:** Changing description to reflect the problem and the corresponding code changes.
    label: documentation
33. Bobby/Daryn, can one of you file tickets for any pending items on this issue? Thanks.
34. Integrated in Hadoop-Yarn-trunk #13 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk/13/]) Updating credits for MAPREDUCE-4229. (Revision 1401493) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401473) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401467) Result = FAILURE daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401493 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401473 Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/StringInterner.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestStringInterner.java daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401467 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/EventReader.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/JobHistoryParser.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/util/CountersStrings.java
35. Integrated in Hadoop-Hdfs-0.23-Build #414 (See [https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/414/]) Updating credits for MAPREDUCE-4229. (Revision 1401487) svn merge -c 1401483 FIXES: MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401485) Result = SUCCESS daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401487 Files : * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401485 Files : * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/StringInterner.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestStringInterner.java * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/EventReader.java * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/JobHistoryParser.java * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/util/CountersStrings.java
36. Integrated in Hadoop-Hdfs-trunk #1205 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk/1205/]) Updating credits for MAPREDUCE-4229. (Revision 1401493) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401473) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401467) Result = FAILURE daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401493 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401473 Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/StringInterner.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestStringInterner.java daryn :

http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401467 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/EventReader.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/JobHistoryParser.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/util/CountersStrings.java

37. Integrated in Hadoop-Mapreduce-trunk #1235 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1235/]) Updating credits for MAPREDUCE-4229. (Revision 1401493) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401473) MAPREDUCE-4229. Intern counter names in the JT (bobby via daryn) (Revision 1401467) Result = SUCCESS daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401493 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401473 Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/StringInterner.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestStringInterner.java daryn : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1401467 Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/EventReader.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/jobhistory/JobHistoryParser.java * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/util/CountersStrings.java