

Item 174

git_comments:

git_commits:

1. **summary:** Add note on HBASE-11118 and HBASE-10877 HBaseZeroCopyByteString woes to troubleshooting section
message: Add note on HBASE-11118 and HBASE-10877 HBaseZeroCopyByteString woes to troubleshooting section

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString"
description: I am running into the problem described in <https://issues.apache.org/jira/browse/HBASE-10304>, while trying to use a newer version within cascading.hbase (<https://github.com/cascading/cascading.hbase>). One of the features of cascading.hbase is that you can use it from lingual (<http://www.cascading.org/projects/lingual/>), our SQL layer for hadoop. lingual has a notion of providers, which are fat jars that we pull down dynamically at runtime. Those jars give users the ability to talk to any system or format from SQL. They are added to the classpath programmatically before we submit jobs to a hadoop cluster. Since lingual does not know upfront, which providers are going to be used in a given run, the HADOOP_CLASSPATH trick proposed in the JIRA above is really clunky and breaks the ease of use we had before. No other provider requires this right now. It would be great to have a programmatical way to fix this, when using fat jars.

jira_issues_comments:

1. Let's look at this again for 0.98.3. [~ndimiduk], [~stack], thoughts?
2. Getting this working for everyone will be tricky. Allow me to ask a silly question: can we build the provider to work around this issue? I don't know how the provider stuff works, but I'm looking at this bit <https://github.com/Cascading/cascading.hbase/blob/2.2/build.gradle#L116-L120>
3. For the provider mechanism: Lingual is a cascading app that submits itself to the cluster. You can use a provider to talk to different data formats/sources. You basically tell lingual in the table definition "this table is actually in Hbase, use this jar file over there to talk to it". lingual itself does not really know about Hbase or any other format/system, but HDFS and delimited files. We create fat jars for those providers to keep the dependency fetching "sane". Only lingual uses those jars. We could def. make them shaded jars, but that will not work in this case, since protobuf is the mechanism to talk to the HBase cluster. Next to that, my understanding of the problem at hand is that it also breaks the classic hadoop jars with lib folders. For those we do not have any control, since our users are just going to use cascading.hbase, build a jar and submit it to the cluster.
4. Can we either: 1. Relocate the protobuf Java runtime library classes somewhere under org.apache.hbase at package time using Maven's shade module (<http://maven.apache.org/plugins/maven-shade-plugin/examples/class-relocation.html>) when producing HBase JARs? If my read of the doc is correct all references in HBase code will be relocated at the bytecode level. 2. Fork the BSD-licensed protobuf Java runtime library classes into a package under org.apache.hbase. ?
5. Class relocation with shade may work. I wonder what it'll break ;)
6. I experimented a bit this morning with shade plugin. Performing the rewrite in hbase-protocol/pom.xml leads to errors building hbase-client. {noformat} [ERROR] /Users/ndimiduk/repos/hbase/hbase-client/src/main/java/org/apache/hadoop/hbase/client/ConnectionManager.java:[1494,33] error: method newBlockingStub in class MasterService cannot be applied to given types; [ERROR] required: com.shaded.google.protobuf.BlockingRpcChannel found: com.google.protobuf.BlockingRpcChannel reason:

- actual argument `com.google.protobuf.BlockingRpcChannel` cannot be converted to `com.shaded.google.protobuf.BlockingRpcChannel` by method invocation conversion `{noformat}` We need the shade plugin to cascade bytecode rewrite down to the dependent modules in the project. I don't see a way to do that.
7. Thanks for trying that out [~ndimiduk]. So then are we down to forking the BSD-licensed protobuf Java runtime library classes into a package under `org.apache.hbase`? Making a blocker for 0.98.3 since it's causing downstreamers grief.
 8. I took the shade question to [user@maven|http://www.mail-archive.com/users@maven.apache.org/msg133791.html]. Maybe some helpful insight will surface.
 9. So do we include the Java protobuf library into our code under the desired (presume eventually shaded) package, as a temporary measure? Need to resolve the blocker and issue for downstreamers. Should proceed with a patch for that.
 10. Still no word on my query to the maven group...
 11. You may want to avert your eyes from the horrors committed in the attached compressed patches for trunk and 0.98. Java sources from protobufs 2.5.0 are imported into `hbase-protocol`, `protobuf-java` deps are removed from POM files (though will still be brought in transitively for some modules, so the POM changes can be left out as meaningless), and the remainder of the changes are made with `{noformat} sed -i -e/com\google\protobuf/org\apache\hadoop\hbase\protobuf/g {noformat}` including generated files (except the descriptor string in `DescriptorProtos`). The changes leak in many places because we use `Service`, generated message builders, and other protobuf classes directly, but it can be done. All unit tests pass.
 12. I hate to push a blocker issue, but I don't see an alternative, moving it to .4. Shading isn't viable thus far, removing HBase's zero copy literal byte string would return the significant performance penalty seen with early versions of 0.96, and the patches I've put up demonstrating how we might incorporate protobuf wholesale are sure to be objectionable. It is my fear however that we will have to do this last thing in order for downstreamers to avoid the classloading issues. I think `cascading.hbase` is already blocked on this.
 13. Pinging the RM crew, [~stack] [~enis] [~lhofhansl]
 14. Thanks for doing this smelly work [~apurtell] and [~ndimiduk]. If we bring in protobuf, how big is that? I count 93 files including tests. That wouldn't be too bad. On plus side, we could make more perf changes.
 15. bq. If we bring in protobuf, how big is that? I rebased on latest trunk. The HBase patch is 293 files changed, 72401 insertions, 24126 deletions. I left out the protobuf tests and only brought over what is in `protobuf-2.5.0/java/src/main/java`, including also the generated `DescriptorProtos.java` from `generated-sources/`. That produces 45 new files in `hbase-protocol`. There are 31 additional test files in `protobuf-2.5.0/java/src/test/`. Simple enough to bring them over into `hbase-protocol` also with substitution. If going this route, we would also need to update the compile-protobuf Maven magic to run a substitution over the generated result with `sed`. I didn't want to spend quality time with Maven if it wasn't necessary. Edit: Fix formatting
 16. One other detail. In the 0.98 patch, because of the `ClusterStatusPublisher/Listener`'s use of a Netty 3 components for marshalling/unmarshalling protobuf, I had to pull in two additional source files from Netty 3 and perform the appropriate substitutions to fix compilation issues. Netty 3 is ASL licensed. I left the RedHat copyright headers alone. This is only necessary for 0.98, trunk is now on Netty 4 and doesn't seem to have the same problem.
 17. I had a quick look at bringing over the tests from `protobuf-2.5.0`. It will be time consuming and involve many substitutions. Probably we can pick a few over that don't represent too much effort and cover the runtime parts.
 18. Comparing against master: <https://github.com/apurtell/hbase/compare/master...HBASE-11118> Comparing against 0.98: <https://github.com/apurtell/hbase/compare/0.98...HBASE-11118-0.98>
 19. So dare we do this for .4 [~stack] [~ndimiduk] ?
 20. Rebased on latest trunk. The GitHub comparison links above will work against the latest.
 21. I was gun-shy at first, but i think this isn't too bad after all. I'd really prefer a shade solution, but not so much so that I'm willing to try hacking it. +1 to forking protobuf. It solves this issue and will allow us to continue make performance improvements as we see fit. Can we bring in the tests as well? How about placing it in it's own module to make this easier to undo down the road? `hbase-protobuf`? Now there's the question of API compatibility. I see our dependency libraries as part of the public API, do others agree? We should not be removing a dependency for a patch release? cc [~nkeywal]
 22. Let's do that. I don't think removing a dependency is an issue: people should not depend on transitive dependencies and anyway it's an easy fix for them imho.
 23. Thanks for publicizing this on dev@ [~ndimiduk]. Let's see what happens.
 24. bq. Can we bring in the tests as well? It's more difficult than the sed based editing I used for the library, because the protobuf for tests specify odd packages deliberately, and also want to test broken cases. Maybe we can bring some over now and more later? bq. How about placing it in it's own module to make this easier to undo down the road? `hbase-protobuf`? We could, but any module depending on one will pull in the other, right? I think it makes sense to put runtime, protos, and generated code into the same module.

25. Have we already considered using our own classloader to isolate our dependency? Forking PB into HBase makes me nervous from a maintenance perspective. PB does new releases once a year or so and they tend to have good improvements - aren't we going to end up stuck on a past version? Instead, could we put some more pressure on the upstream protobuf maintainers to include the ZeroCopyLiteralByteString, or at least make its super-class non-final in order to support this? Alternatively, could we get shading to work by adding an extra indirection package that builds a "jar-with-dependencies" of both hbase-protocol and protobuf, and then shades that and re-publishes as an "hbase-protocol-with-shaded-pb" pom? Then the rest of HBase could depend on that?
26. bq. Instead, could we put some more pressure on the upstream protobuf maintainers to include the ZeroCopyLiteralByteString, or at least make its super-class non-final in order to support this? I think Stack had a conversation without a positive result. bq. Alternatively, could we get shading to work by adding an extra indirection package that builds a "jar-with-dependencies" of both hbase-protocol and protobuf, and then shades that and re-publishes as an "hbase-protocol-with-shaded-pb" pom? Then the rest of HBase could depend on that? Perhaps. I think I don't have the necessary Maven expertise to pull that off because I can't envision how to accomplish that.
27. I'm also not sure upstream changes to protobuf can help HBase 0.98 / Hadoop 2 users.
28. Has using an alternative protobuf java lib like protostuff been considered? With that we could likely have more influence and get performance-oriented changes in and eventually percolate it to other ecosystem projects.
29. No. Sounds interesting. Do we have a volunteer to replace our use of protobuf with protostuff (or whatever) and benchmark the performance delta under load on a cluster? Not sure I can find the time for that. Meanwhile some downstreamers like cascading are not able to use 0.98 or recent 0.96.
30. How do we sync our use of google's protoc in the compile-protobuf build target with an alternative runtime? I'm not familiar with protostuff. Does it have its own compiler?
31. [~tlipton] bq. Have we already considered using our own classloader to isolate our dependency? There is a bit of messy research done at the head of HBASE-10304. IIRC, we don't get chance to do interception. RunJar is running the show. That is the case here too [~fs111]? Are you using hadoop runjar or a similar mechanism? bq. Instead, could we put some more pressure on the upstream protobuf maintainers to include the ZeroCopyLiteralByteString, or at least make its super-class non-final in order to support this? <https://code.google.com/p/protobuf/issues/detail?id=374> asks and gets shutdown because it would break base precept that all is immutable in pb. bq. Alternatively, could we get shading to work by adding an extra indirection package that builds a "jar-with-dependencies" of both hbase-protocol and protobuf, and then shades that and re-publishes as an "hbase-protocol-with-shaded-pb" pom? Then the rest of HBase could depend on that? Let me try it Todd.
32. I was looking at protostuff last night. Seems dead but for a few recent commits and no mention of pb 2.5. [~ryanobjc] or [~posix4e]? Any input?
33. Protostuff is wireline compatible with protobuf 2.5. The project is in active use, and we maintain, improve and fix it. In terms of API, it has a different API than protobuf, by necessity, since ByteString, by design, forces a copy on you. Our solution is to use ByteBuffer as an alternative fundamental byte-holder type. Additionally, fixing and improving the project is really trivial. The code is easy to work with, it's fully in Java (including the compiler), the code-generation is a templating engine, and no more committing generated code (big anti-pattern).
34. piling on Daniel Yu, the project creator is definitely still very responsible and involved. He has been very helpful and easy to work with. We recently got some pull requests from some new contributors which is very exciting. We intend on merging them soon. OhmData is about to push some patches to further improve read performance, expect them soonish. After those hit, we will cut a new release.
35. Protostuff sounds really interesting
36. [~todd@lipcon.org] Cascading uses the JobClient to submit the jobs on the fly, but we don't call the hadoop shell wrapper for each job, like hive does. All submission is handled internally. The initial program gets started with hadoop/yarn jar myJar, and then we handle everything ourselves. In the lingual case, it is even more complicated, since we fetch jars dynamically at runtime and add them to the classpath of the jobs, which means we cannot know upfront, if the job is going to use cascading-hbase or not.
37. I looked at protostuff a bit more. According to ohloh.net, {quote} In a Nutshell, protostuff... has had 1,090 commits made by 7 contributors representing 68,359 lines of code ... is mostly written in Java with a low number of source code comments ... has a codebase with a long source history maintained by by one developer with stable Y-O-Y commits {quote} This might be all from before the fork from Google Code to GitHub. The GitHub project has 3 contributors. It's nice the compiler is Java. There is an available Maven build plugin. Extensions are not supported (but we don't use them) We will need to support any coprocessor implementor who coded an Endpoint, using the com.google.protobuf API. This means adding classes in a com.google.protobuf package that forward to the real implementation. The current patch also has this issue. The API is quite different from protobuf-java 2.5 and missing all of the RpcChannel and Service bits. We will

- need to reimplement our RPC one way or another, probably by bringing over RpcChannel and Service from protobuf and implementing them on top of protostuff if we want to be API compatible with coprocessors as above and our existing codebase. All this for a ByteString that doesn't copy...
38. My comment was lost. bq. This means adding classes in a com.google.protobuf package that forward to the real implementation. Will this put us back in the same spot? i.e. we'll load the wrong c.g.p classes? bq. ...and missing all of the RpcChannel and Service bits. Bad. This will be bunch of work.
39. {quote} bq. This means adding classes in a com.google.protobuf package that forward to the real implementation. Will this put us back in the same spot? i.e. we'll load the wrong c.g.p classes? {quote} Good point. Well, there will be a compatibility break then, for coprocessors. Best make it for 0.99+. Let me remove 0.98 from the target versions for now. We can put it back if we figure out something there.
40. Thinking on it more, not generating the rpc/server bindings is a blocker. It would be mountains of work hand coding something that is now generated hooking up rpc invocation w/ Service call through to server/client... unless [~ryanobjc] or [~posix4e] you fellows have a patch? Let me attach a suggestion. It moves HBaseZeroCopyLiteralByteString into hbase-server so it is out of the way of client dependencies. Means we have the speedup serverside only.
41. And it should be out of the way of mr dependencies....
42. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12650953/1118.suggested.undoing.optimization.on.clientside.txt> against trunk revision . ATTACHMENT ID: 12650953 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified tests. {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:red}-1 findbugs{color}. The patch appears to introduce 1 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 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: {color:red}-1 core zombie tests{color}. There are 1 zombie test(s): at org.apache.hadoop.hbase.regionserver.TestHRegion.testWritesWhileGetting(TestHRegion.java:3499) Test results: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-thrift.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html> Console output: <https://builds.apache.org/job/PreCommit-HBASE-Build/9787//console> This message is automatically generated.
43. Retry. The failure seems unrelated. The findbugs is not mine.
44. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12651183/1118.suggested.undoing.optimization.on.clientside.txt> against trunk revision . ATTACHMENT ID: 12651183 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified tests. {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:red}-1 findbugs{color}. The patch appears to introduce 1 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 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: {color:red}-1 core zombie tests{color}. There are 1 zombie test(s): Test

results: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-thrift.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html> Console output: <https://builds.apache.org/job/PreCommit-HBASE-Build/9790//console> This message is automatically generated.

45. Sounds like you found the right solution. Glad we could help give some perspective on protostuff.
46. bq. Thinking on it more, not generating the rpc/server bindings is a blocker. It would be mountains of work hand coding something that is now generated hooking up rpc invocation w/ Service call through to server/client Agreed. bq. It moves HBaseZeroCopyLiteralByteString into hbase-server so it is out of the way of client dependencies. Means we have the speedup serverside only. bq. And it should be out of the way of mr dependencies.... We do not have a module for mapreduce. You mean the MR code should not import the class, hence it won't be loaded. Sounds good. Can we do a util class, which checks whether it can load up ZCLBS and uses it if so, otherwise defaults to BS.copyFrom()?
47. bq. You mean the MR code should not import the class, hence it won't be loaded. I mean these fat jars that want to talk to hbase won't be bundling hbase-server (unless by mistake) so they shouldn't be trying to load HBaseZeroCopyLiteralString. But maybe I need to do more pruning and get it out of these MR utilities too, at least I'd have to fix <https://issues.apache.org/jira/browse/HBASE-10304> [~fs111] What exception trace are you seeing? Is it same as HBASE-10304? i.e. Lingual wants to run mapreduce job? It is making use of hbase mapreduce formats? bq. Can we do a util class, which checks whether it can load up ZCLBS and uses it if so, otherwise defaults to BS.copyFrom() Let me see. Maybe if it loads before everything else...
48. -- protostuff dev here Reading from your comments, I think the most logical choice is to fork protobuf and override the ByteString behavior to wrap instead of copy (either modify ByteString.copyFrom or modify the protoc-codegen to emit ByteString.wrap). Maintenance is easy with upstream since you're only modifying one source (ByteString) which will not be evolving much at all. "Thinking on it more, not generating the rpc/server bindings is a blocker. It would be mountains of work hand coding something that is now generated hooking up rpc invocation w/ Service call through to server/client" If you still want to experiment with protostuff, I can help with this. Its only a matter of having existing code, then converting that into a template. Looking at the codebase, is it safe to assume that hbase's use of protobuf-rpc are all blocking (not asynchronous)? Do you use the reflection bits at all (e.g newReflectiveService)? I see some hacks like "Ugly delegation just so we can add in a Close method.", which I guess is caused by the inability to customize the codegen. [~posix4e] Daniel aye?
49. [~d4niel] Thank you for taking the time to stop by. bq. Reading from your comments, I think the most logical choice is to fork protobuf and override the ByteString behavior to wrap instead of copy (either modify ByteString.copyFrom or modify the protoc-codegen to emit ByteString.wrap). The problem is classloading a protobuf ahead of our doctored one, one that is in the CLASPATH already put there by hadoop. bq. Its only a matter of having existing code, then converting that into a template. Thank you. What happens then if we add new methods to the Interfaces? We'd have to modify the template? (The protobuf generated classes are massive, they are half our lines of code). bq. Looking at the codebase, is it safe to assume that hbase's use of protobuf-rpc are all blocking... You are correct. We want to move to async but its a bit of work that we've put off for the moment. bq. which I guess is caused by the inability to customize the codegen. Yeah, I suppose we're being lazy and we should be going back modifying protoc generation (that is one of the attractive things about protostuff... that we could add it into our inline build along w/ any mods we'd want to package). Again, thanks for coming by Daniel.
50. First off, I'm not Daniel :-) {quote} The problem is classloading a protobuf ahead of our doctored one, one that is in the CLASPATH already put there by hadoop. {quote} I see. Would hadoop consider moving to your

- forked protobuf? If adding `ByteString.wrap` is the only modification, along with a field option (`wrap = true`) to make the protoc codegen emit `ByteString.wrap`, I'm sure its not a problem since no incompatibility will arise (new behavior is only activated by a new field option). {quote} Thank you. What happens then if we add new methods to the Interfaces? We'd have to modify the template? {quote} No modification needed. You'll just have to use a service option/annotation that activates the extra method generation in the template. {quote} (The protobuf generated classes are massive, they are half our lines of code). {quote} Indeed they are. There are variants (`LITE_RUNTIME`, etc) when you do not need the extra stuff (reflection, extension, descriptor, unknown fields, etc). Ofc you can always customize the output by working with protoc-gen {quote} Yeah, I suppose we're being lazy and we should be going back modifying protoc generation {quote} Yes and yes
51. [~stack] : We do have the hbase-server deps in the jars. This is due to the fact that we are based on a fork of the mapred InputFormat/OutputFormat, which uses the TableSplit class to calculate the InputSplits. The problem is that we cannot use the mapreduce apis in cascading and our version of the Input and OutputFormat has evolved and we want to keep it for backwards compatibility. If there is a way to get this working w/o the hbase-server deps, I am more than happy to drop it. The code is here, if that helps:
<https://github.com/Cascading/cascading.hbase>
52. bq. First off, I'm not Daniel Sorry [~david.yu]. I messed up. bq. Would hadoop consider moving to your forked protobuf? This would work for future hadoops but not for what is currently out in the world (folks tend to mix and match what version runs on what). Besides, by the time it was widely available, we'll be up on protostuff so won't matter (smile). [~fs111] Ok. Thanks. So, the exception stack is same as over in HBASE-10304? You are coming in via TableMapReduceUtil (from mapred package?) Thanks.
53. This seems to work. Verified using Nicks nice <https://github.com/ndimiduk/hbase-fatjar> project. Adds new ByteString class that on class loading tries to make use of HBaseZeroCopyByteString. If fails to load -- IllegalAccessError -- then we set flag and everafter, do `ByteString.copyFrom` instead. Patch is big because lots of HBZCBS replacements. My other patch moving HBZCBS to hbase-server wouldn't have worked given mr classes are in hbase-server (duh!) and these are the gateway used that brings on this problem (need to work on breaking out the hbase-mapreduce module).
54. [~fs111] Any chance of your trying out this last patch? If too much work for you to do, just say and I'll borrow the head of 0.98, apply this patch temporarily and make you a SNAPSHOT to try.
55. bq. Adds new ByteString class that on class loading tries to make use of HBaseZeroCopyByteString. If fails to load – IllegalAccessError – then we set flag and everafter, do `ByteString.copyFrom` instead. This sounds like a good solution for 0.98.
56. For future reference, there are three post-protobuf options for consideration (at least):
<https://kentonv.github.io/capnproto/news/2014-06-17-capnproto-flatbuffers-sbe.html>
57. [~apurtell] None are pb compat on quick read? (Its not mentioned in the matrix...)
58. Does it make sense to have a non-protobuf dependent RPC framework? At this point isn't protobuf not just 'turn structs into bytes' but also deeply entangled in to HBase? Seems like a longer term project.
59. No, none are PB compatible. Can't be if meeting the desired goal of eliminating serde and copying costs. As a longer term project absolutely. Making a note here.
60. [~stack] I have tried getting this to work. I took the master, applied the patch, created a tarball, as explained here: <http://hbase.apache.org/book/releasing.html#maven.release> and tried to fire up a cluster. The regionservers come up cleanly, but the master is not available on <http://host:60010>. I can't find anything in the logs. Is there something I overlooked?
61. [~stack] I figured out how where the master web ui is now, but I fail to put working jars in my local maven repo. I can't find a way to resolve `_${compat.module}` correctly. What is the exact invocation I need to get this working?
62. master has dropped support for hadoop-1. Can you provide the commands you've used and the error you're seeing? Thanks.
63. [~fs111] Tell us what to do so testing is easy for you? I could branch 0.98, commit this patch, then push a build to mvn if that would help you.
64. Here is what I did: `mvn versions:set -DnewVersion=0.99-Cascading mvn install -DskipTests site assembly:single -Prelease` I then took the binary tarball to deploy a cluster. Now if I want to rebuild the cascading.hbase module, it always fails with an error related to the `compat.module`. It seems the variable hasn't been expanded and that confuses gradle:
65. Sorry, forgot the gradle error: {code} \$ gradle cascading-hbase-hadoop2-mr1:build The TaskContainer.add() method has been deprecated and is scheduled to be removed in Gradle 2.0. Please use the create() method instead. :cascading-hbase-hadoop2-mr1:compileJava FAILURE: Build failed with an exception. * What went wrong: Could not resolve all dependencies for configuration ':cascading-hbase-hadoop2-mr1:compile'. > Could not resolve org.apache.hbase:\$_{compat.module}:0.99-CASCADING. Required by: cascading-hbase:cascading-hbase-hadoop2-mr1:2.5.0-wip-dev > org.apache.hbase:hbase-server:0.99-CASCADING cascading-hbase:cascading-hbase-hadoop2-mr1:2.5.0-wip-dev > org.apache.hbase:hbase-server:0.99-CASCADING > org.apache.hbase:hbase-prefix-tree:0.99-CASCADING > java.lang.NullPointerException

- (no error message) > java.lang.IllegalArgumentException (no error message) >
java.lang.IllegalArgumentException (no error message) > java.lang.IllegalArgumentException (no error message) * Try: Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. BUILD FAILED Total time: 8.848 secs {code}
66. [~fs111] Can you pass a property? compat.module=hbase-hadoop2-compat ? Or hard-code it in the pom?
67. +1 for the ByteString patch. It seems we do not need to do reflection on the HZCLBS, so that we can import that class, right?
68. bq. It seems we do not need to do reflection on the HZCLBS, so that we can import that class, right? Not for it, no. I was hoping to get some confirmation from [~fs111] that this patch works for him but I could commit (I reproduced using Nick's fatjar project and this patches fixes the issue in that case).
69. I think we can go ahead and commit this to trunk and 0.98
70. Sorry for not coming back to this. I haven't been able to test this, since I am busy with more important matters. If you could point me to a snapshot build, that I could use, I can give this a spin. I seem to be unable to build a custom HBase.
71. I published a SNAPSHOT for you [~fs111] up at
<https://repository.apache.org/content/repositories/snapshots/org/apache/hbase/> SNAPSHOT is named: 0.98.4-11118-SNAPSHOT Hopefully you get a chance to test. Thanks.
72. Patch for 0.98.
73. {color:red}-1 overall{color}. Here are the results of testing the latest attachment
<http://issues.apache.org/jira/secure/attachment/12653477/11118.098.txt> against trunk revision .
ATTACHMENT ID: 12653477 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 21 new or modified tests.
{color:red}-1 patch{color}. The patch command could not apply the patch. Console output:
<https://builds.apache.org/job/PreCommit-HBASE-Build/9930//console> This message is automatically generated.
74. [~stack] Thanks for the snapshot. I hope I will find a window of time to give it a spin in the next few days. I will post results here.
75. Hi [~stack]. Taking 11118.098.txt for a spin. I applied the patch to 0.98 and launched local mode. I'm seeing this in my log {noformat} 2014-07-02 13:24:53,899 ERROR [htable-pool9-t1] client.AsyncProcess: #6, Caught throwable while calling. This is unexpected. Retrying. Server is 10.11.5.156,54722,1404332679793, tableName=hbase:meta java.lang.RuntimeException: java.lang.StackOverflowError at org.apache.hadoop.hbase.client.RpcRetryingCaller.callWithoutRetries(RpcRetryingCaller.java:188) at org.apache.hadoop.hbase.client.AsyncProcess\$1.run(AsyncProcess.java:543) at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:471) at java.util.concurrent.FutureTask\$Sync.innerRun(FutureTask.java:334) at java.util.concurrent.FutureTask.run(FutureTask.java:166) 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:724) Caused by: java.lang.StackOverflowError at org.apache.hadoop.hbase.util.ByteString.wrap(ByteString.java:65) at org.apache.hadoop.hbase.util.ByteString.wrap(ByteString.java:65) at org.apache.hadoop.hbase.util.ByteString.wrap(ByteString.java:65) ... {noformat} Attaching a fixed patch, simple interdiff: {noformat} diff -u b/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java b/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java --- b/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java +++ b/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java @@ -62,7 +62,7 @@ * Wraps a subset of a byte array in a {@link ByteString} without copying it. */ public static ByteString wrap(final byte[] array, int offset, int length) { - return USE_ZEROCOPYBYTESTRING? ByteString.wrap(array, offset, length): + return USE_ZEROCOPYBYTESTRING? HBaseZeroCopyByteString.wrap(array, offset, length): ByteString.copyFrom(array, offset, length); } } {noformat} I also updated the hbase-fatjar to include the apache repo to make it easier to test snapshots.
76. {color:red}-1 overall{color}. Here are the results of testing the latest attachment
<http://issues.apache.org/jira/secure/attachment/12653686/11118.098-0.txt> against trunk revision .
ATTACHMENT ID: 12653686 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 21 new or modified tests.
{color:red}-1 patch{color}. The patch command could not apply the patch. Console output:
<https://builds.apache.org/job/PreCommit-HBASE-Build/9946//console> This message is automatically generated.
77. Thanks [~ndimiduk] [~fs111] I refreshed the snapshot to include fix that Nick posted. Just FYI.

78. [~stack] FYI, building my fat-jar project with -Dhbase.version=0.98.4-11118-SNAPSHOT isn't pulling the snapshot jars from the apache repo. I don't know if it's something wrong with that project's pom or with the snapshot release itself.
79. You patch includes the necessary changes to pom.xml files, excellent! I installed this build and ran the fat-jar tests. This patch fixes the issue, at least as far as this test case is concerned. Nice one [~stack]!
80. Attaching patches for 0.98 and master. Please consider these candidates for commit. These are basically just [~stack]'s patches, rebased to appropriate HEAD and omitting the pom changes.
81. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12654375/HBASE-11118.00.patch> against trunk revision . ATTACHMENT ID: 12654375 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 28 new or modified tests. {color:red}-1 javac{color}. The patch appears to cause mvn compile goal to fail. Compilation errors resume: [ERROR] COMPILATION ERROR : [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[20,33] error: package org.apache.commons.logging does not exist [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[21,33] error: package org.apache.commons.logging does not exist [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[31,23] error: cannot find symbol [ERROR] symbol: class Log [ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:2.5.1:compile (default-compile) on project hbase-protocol: Compilation failure: Compilation failure: [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[20,33] error: package org.apache.commons.logging does not exist [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[21,33] error: package org.apache.commons.logging does not exist [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[31,23] error: cannot find symbol [ERROR] symbol: class Log [ERROR] location: class ByteString [ERROR] /home/jenkins/jenkins-slave/workspace/PreCommit-HBASE-Build/hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java:[31,33] error: cannot find symbol [ERROR] -> [Help 1] [ERROR] [ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch. [ERROR] Re-run Maven using the -X switch to enable full debug logging. [ERROR] [ERROR] For more information about the errors and possible solutions, please read the following articles: [ERROR] [Help 1] <http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException> [ERROR] [ERROR] After correcting the problems, you can resume the build with the command [ERROR] mvn <goals> -rf :hbase-protocol Console output: <https://builds.apache.org/job/PreCommit-HBASE-Build/9983/console> This message is automatically generated.
82. +1 for 0.98. Please fix imports on commit
83. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12654382/HBASE-11118.01.patch> against trunk revision . ATTACHMENT ID: 12654382 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 28 new or modified tests. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {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:red}-1 findbugs{color}. The patch appears to introduce 4 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 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: org.apache.hadoop.hbase.client.TestMultiParallel Test results: <https://builds.apache.org/job/PreCommit-HBASE-Build/9985/testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9985/artifact/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9985/artifact/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9985/artifact/patchprocess/newPatchFindbugsWarningshbase-client.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9985/artifact/patchprocess/newPatchFindbugsWarningshbase-common.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9985/artifact/patchprocess/newPatchFindbugsWarningshbase-common.html>

- Build/9985//artifact/patchprocess/newPatchFindbugsWarningshbase-protocol.html Findbugs warnings:
<https://builds.apache.org/job/PreCommit-HBASE-Build/9985//artifact/patchprocess/newPatchFindbugsWarningshbase-server.html> Findbugs warnings:
<https://builds.apache.org/job/PreCommit-HBASE-Build/9985//artifact/patchprocess/newPatchFindbugsWarningshbase-examples.html> Findbugs warnings:
<https://builds.apache.org/job/PreCommit-HBASE-Build/9985//artifact/patchprocess/newPatchFindbugsWarningshbase-thrift.html> Findbugs warnings:
<https://builds.apache.org/job/PreCommit-HBASE-Build/9985//console> This message is automatically generated.
84. +1 for branch-1.
85. bq. Please fix imports on commit I went through all files touched by both patches, removing unused imports. Attaching the results as v02. Please let me know if something else was intended.
86. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12654444/HBASE-11118.02.patch> against trunk revision . ATTACHMENT ID: 12654444 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 28 new or modified tests. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {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:red}-1 findbugs{color}. The patch appears to introduce 4 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 lineLengths{color}. The patch does not introduce lines longer than 100 {color:green}+1 site{color}. The mvn site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: org.apache.hadoop.hbase.coprocessor.TestRegionObserverInterface Test results: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-hadoop2-compat.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-client.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-common.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-protocol.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-server.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-examples.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//artifact/patchprocess/newPatchFindbugsWarningshbase-thrift.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HBASE-Build/9988//console> This message is automatically generated.
87. +1
88. Really happy we didn't have to pull the trigger on forking protobuf after all, thanks [~stack]
89. Hi [~fs111], we have a solution. It would be great if you could verify it works for you before we pull the trigger and declare success. Thanks.
90. [~ndimiduk] no luck with the snapshot repository: \$ gradle clean build The TaskContainer.add() method has been deprecated and is scheduled to be removed in Gradle 2.0. Please use the create() method instead. :cascading-hbase-hadoop2-mr1:clean UP-TO-DATE :cascading-hbase-hadoop2-mr1:compileJava Download <https://repository.apache.org/content/repositories/snapshots/org/apache/hbase/hbase-hadoop2-compat/0.98.4-11118-SNAPSHOT/hbase-hadoop2-compat-0.98.4-11118-20140702.231131-2.pom> FAILURE: Build failed with an exception. * What went wrong: Could not resolve all dependencies for configuration ':cascading-hbase-hadoop2-mr1:compile'. > Could not resolve org.apache.hbase:\${compat.module}:0.98.4-11118-SNAPSHOT. Required by: cascading-hbase:cascading-hbase-hadoop2-mr1:2.5.0-wip-dev > org.apache.hbase:hbase-server:0.98.4-11118-SNAPSHOT cascading-hbase:cascading-hbase-hadoop2-mr1:2.5.0-wip-dev > org.apache.hbase:hbase-server:0.98.4-11118-SNAPSHOT > org.apache.hbase:hbase-

prefix-tree:0.98.4-11118-SNAPSHOT > java.lang.NullPointerException (no error message) > Unable to load Maven meta-data from [http://repo1.maven.org/maven2/org/apache/hbase/\\${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml](http://repo1.maven.org/maven2/org/apache/hbase/${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml). > java.lang.IllegalArgumentException (no error message) > Unable to load Maven meta-data from [http://conjars.org/repo/org/apache/hbase/\\${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml](http://conjars.org/repo/org/apache/hbase/${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml). > java.lang.IllegalArgumentException (no error message) > Unable to load Maven meta-data from [https://repository.apache.org/content/repositories/releases/org/apache/hbase/\\${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml](https://repository.apache.org/content/repositories/releases/org/apache/hbase/${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml). > java.lang.IllegalArgumentException (no error message) > Unable to load Maven meta-data from [https://repository.apache.org/content/repositories/snapshots/org/apache/hbase/\\${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml](https://repository.apache.org/content/repositories/snapshots/org/apache/hbase/${compat.module}/0.98.4-11118-SNAPSHOT/maven-metadata.xml). > java.lang.IllegalArgumentException (no error message) * Try: Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. BUILD FAILED Total time: 18.433 secs

91. Hi [~fs111]. I think I've reproduced the repo issue, let me see about fixing it. In the mean time, I've pushed a branch to my github. Can you test against a local repo? If so, checkout the branch, `mvn clean install -DskipTests` will install an artifact version 0.98.4-SNAPSHOT built against hadoop2. Can you give that a spin instead? Thanks. <https://github.com/ndimiduk/hbase/tree/11118-0.98-ZeroCopyLiteralByteString-fix>
92. Hi [~fs111]. I've "published" a built of this branch. Add the following repository info and give it another spin. Thanks {noformat} <repository> <id>apache.people.ndimiduk</id>
<url><http://people.apache.org/~ndimiduk/repository></url> <snapshots> <enabled>true</enabled>
</snapshots> </repository> {noformat}
93. Same as before, artifact version is still 0.98.4-SNAPSHOT, built against hadoop2.
94. This issue seems stuck on Maven/repo madness. We think we have a patch that fixes the issue, it has a +1, so I am going to commit this tomorrow so we can make forward progress and get out a 0.98.4 release, unless someone vetoes between now and then. Confirmation would be great but the snapshot repos have not been working out. We can verify through normal Maven channels with .4.
95. Precisely my plan. I'll take care of it if I see you haven't. Don't want to hold up the RC any longer.
96. [~ndimiduk] I tried your repo and installing it myself locally, but none of it works. The compat.module variable is never expanded and breaks everything.
97. [~fs111] I pulled cascading.hbase and did a {{gradle clean build}}. I think the reason you're seeing compat.module issues is because this project has dependencies against hbase/hadoop1 and I only pushed hadoop2 artifacts. {noformat} :check UP-TO-DATE :build :cascading-hbase-hadoop:compileJava Download <http://repo1.maven.org/maven2/org/apache/hbase/hbase-client/0.98.1-hadoop1/hbase-client-0.98.1-hadoop1.pom> Download <http://repo1.maven.org/maven2/org/apache/hbase/hbase/0.98.1-hadoop1/hbase-0.98.1-hadoop1.pom> Download <http://repo1.maven.org/maven2/org/apache/hbase/hbase-server/0.98.1-hadoop1/hbase-server-0.98.1-hadoop1.pom> Download <http://repo1.maven.org/maven2/org/apache/hbase/hbase-common/0.98.1-hadoop1/hbase-common-0.98.1-hadoop1.pom> Download <http://repo1.maven.org/maven2/org/apache/hbase/hbase-hadoop1-compat/0.98.1-hadoop1/hbase-hadoop1-compat-0.98.1-hadoop1.pom> ... {noformat} Let me build both hadoop compat modules and give this a try.
98. yes, but I was doing my tests in cascading-hbase-hadoop2-mr1, which is only using hadoop2.
99. [~fs111] please give it another shot. The compat.module issues are a side-effect of how the HBase build is managed for hadoop1/hadoop2 compatibility. Basically, Stack and I were both lazy in how we built and published jars. I've done it properly now and the build of cascading.hbase is now working for me with the following patch applied. {noformat} diff --git a/build.gradle b/build.gradle index 75f3b46..5f25ea9 100644 --
- a/build.gradle +++ b/build.gradle @@ -60,6 +60,7 @@ allprojects { mavenCentral() maven{ url
'http://conjars.org/repo/' } maven{ url 'https://repository.apache.org/content/repositories/releases/' } + maven{
url 'http://people.apache.org/~ndimiduk/repository/' } } version = releaseVersion } diff --git a/cascading-
hbase-hadoop/build.gradle b/cascading-hbase-hadoop/build.gradle index 6507a1a..d631762 100644 ---
a/cascading-hbase-hadoop/build.gradle +++ b/cascading-hbase-hadoop/build.gradle @@ -20,7 +20,7 @@
apply from: "\${rootDir}/etc/hadoop-shared-config.gradle" -ext.hbaseVersion = '0.98.1-hadoop1'
+ext.hbaseVersion = '0.98.4-hadoop1-SNAPSHOT' ext.hadoopVersion = '1.2.1' dependencies { diff --git
a/cascading-hbase-hadoop2-mr1/build.gradle b/cascading-hbase-hadoop2-mr1/build.gradle index
f36fc99..91e62ad 100644 --- a/cascading-hbase-hadoop2-mr1/build.gradle +++ b/cascading-hbase-hadoop2-
mr1/build.gradle @@ -24,7 +24,7 @@ idea { pathVariables MODULE_DIR: file("\${rootDir}/cascading-
hbase-hadoop") } -ext.hbaseVersion = '0.98.1-hadoop2' +ext.hbaseVersion = '0.98.4-hadoop2-SNAPSHOT'
ext.hadoopVersion = '2.2.0' dependencies { {noformat}
100. Ok [~ndimiduk]
101. Patch for 0.98 needed slit update after rebase, uploading what was committed.
102. I have it on good authority that [~fs111] has finished work for the day, so committing what's here. Please re-open for addendum or open a new ticket if this issue persists. Pushed to 0.98, branch-1, and master.

103. Just realized I pushed a commit message that neglected to honor [~stack] as the contribution's primary author. My apologies sir!
104. FAILURE: Integrated in HBase-0.98 #389 (See [<https://builds.apache.org/job/HBase-0.98/389/>]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString" (ndimiduk: rev 8884ad0444cada23dfbd4dbde9db9e2a20ae262a) * hbase-server/src/test/java/org/apache/hadoop/hbase/regionserver/TestPriorityRpc.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityUtils.java * hbase-server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestCoproprocessorEndpoint.java * hbase-server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestRowProcessorEndpoint.java * hbase-server/src/main/java/org/apache/hadoop/hbase/protobuf/ReplicationProtobufUtil.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FirstKeyValueMatchingQualifiersFilter.java * hbase-protocol/src/main/java/org/apache/hadoop/hbase/util/ByteStringifier.java * hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java * hbase-server/src/test/java/org/apache/hadoop/hbase/protobuf/TestProtobufUtil.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/InclusiveStopFilter.java * hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/HRegionServer.java * hbase-server/src/main/java/org/apache/hadoop/hbase/rest/model/CellSetModel.java * hbase-client/src/main/java/org/apache/hadoop/hbase/security/access/AccessControlClient.java * hbase-client/src/test/java/org/apache/hadoop/hbase/client/TestClientNoCluster.java * hbase-client/src/main/java/org/apache/hadoop/hbase/HTableDescriptor.java * hbase-client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/SecureBulkLoadClient.java * hbase-server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/ColumnRangeFilter.java * hbase-server/src/main/java/org/apache/hadoop/hbase/codec/MessageCodec.java * hbase-server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFile.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/ByteArrayComparable.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPrefixFilter.java * hbase-server/src/main/java/org/apache/hadoop/hbase/rest/model/CellModel.java * hbase-server/src/main/java/org/apache/hadoop/hbase/io/Reference.java * hbase-client/src/main/java/org/apache/hadoop/hbase/security/EncryptionUtil.java * hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/RequestConverter.java * hbase-server/src/main/java/org/apache/hadoop/hbase/rest/model/ScannerModel.java * hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/HLogKey.java * hbase-client/src/main/java/org/apache/hadoop/hbase/ClusterStatus.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/DependentColumnFilter.java * hbase-protocol/pom.xml * hbase-client/src/main/java/org/apache/hadoop/hbase/ipc/RegionCoproprocessorRpcChannel.java * hbase-client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/BigDecimalColumnInterpreter.java * hbase-server/src/test/java/org/apache/hadoop/hbase/replication/regionserver/TestReplicationSink.java * hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/SecureProtobufLogWriter.java * hbase-client/src/main/java/org/apache/hadoop/hbase/ipc/MasterCoproprocessorRpcChannel.java * hbase-client/src/main/java/org/apache/hadoop/hbase/HRegionInfo.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/SingleColumnValueFilter.java * hbase-server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFileBlock.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/MultipleColumnPrefixFilter.java * hbase-client/src/main/java/org/apache/hadoop/hbase/HColumnDescriptor.java * hbase-server/src/main/java/org/apache/hadoop/hbase/io/hfile/FixedFileTrailer.java * hbase-protocol/src/main/java/com/google/protobuf/HBaseZeroCopyByteString.java * hbase-client/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityClient.java * hbase-server/src/main/java/org/apache/hadoop/hbase/rest/model/TableInfoModel.java * hbase-server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestBatchCoproprocessorEndpoint.java * hbase-client/src/main/java/org/apache/hadoop/hbase/RegionTransition.java * hbase-client/src/main/java/org/apache/hadoop/hbase/zookeeper/ZKUtil.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/PrefixFilter.java * hbase-server/src/main/java/org/apache/hadoop/hbase/rest/model/StorageClusterStatusModel.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPaginationFilter.java
105. FAILURE: Integrated in HBase-TRUNK #5290 (See [<https://builds.apache.org/job/HBase-TRUNK/5290/>]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString" (ndimiduk: rev c01e7e50c1047565da1b0c6ed6fe8bfdf25e03b4) *

hbase-server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/SecureProtobufLogWriter.java *
 hbase-server/src/main/java/org/apache/hadoop/hbase/rest/model/TableInfoModel.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/HLogKey.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestRowProcessorEndpoint.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFileBlock.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/ipc/RegionCoprocessorRpcChannel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/BigDecimalColumnInterpreter.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/replication/regionserver/TestReplicationSink.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/security/access/AccessControlClient.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/FirstKeyValueMatchingQualifiersFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/ClusterStatus.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ByteArrayComparable.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPaginationFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/PrefixFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ColumnRangeFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/HTableDescriptor.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/codec/MessageCodec.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/FixedFileTrailer.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPrefixFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/protobuf/ReplicationProtobufUtil.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/SecureBulkLoadClient.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/DependentColumnFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/ScannerModel.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/protobuf/TestProtobufUtil.java * hbase-
 protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFile.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/CellSetModel.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/regionserver/HRegionServer.java * hbase-
 client/src/test/java/org/apache/hadoop/hbase/client/TestClientNoCluster.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/HRegionInfo.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/InclusiveStopFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/ipc/MasterCoprocessorRpcChannel.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/StorageClusterStatusModel.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/Reference.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/regionserver/TestPriorityRpc.java * hbase-
 protocol/src/main/java/com/google/protobuf/HBaseZeroCopyByteString.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/zookeeper/ZKUtil.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityUtils.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/RegionTransition.java * hbase-protocol/pom.xml * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/SingleColumnValueFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/CellModel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityClient.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestBatchCoprocessorEndpoint.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/protobuf/RequestConverter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/HColumnDescriptor.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestCoprocessorEndpoint.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/security/EncryptionUtil.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/MultipleColumnPrefixFilter.java

106. FAILURE: Integrated in HBase-1.0 #31 (See [<https://builds.apache.org/job/HBase-1.0/31/>]) HBASE-11118
 non environment variable solution for "IllegalAccessError: class
 com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass
 com.google.protobuf.LiteralByteString" (ndimiduk: rev c3628579013d7f5e1dbdfdf0b9ceca0221c1c1e66) *
 hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPrefixFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/FixedFileTrailer.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/RegionTransition.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityUtils.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/FirstKeyValueMatchingQualifiersFilter.java * hbase-

server/src/main/java/org/apache/hadoop/hbase/protobuf/ReplicationProtobufUtil.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/PrefixFilter.java * hbase-
protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/regionserver/TestPriorityRpc.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/SecureProtobufLogWriter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityClient.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/rest/model/CellModel.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/security/access/AccessControlClient.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/security/EncryptionUtil.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFileBlock.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/codec/MessageCodec.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/ipc/RegionCoprocessorRpcChannel.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/regionserver/HRegionServer.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/zookeeper/ZKUtil.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/HLogKey.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFile.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/protobuf/RequestConverter.java * hbase-protocol/pom.xml *
hbase-client/src/main/java/org/apache/hadoop/hbase/filter/ByteArrayComparable.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/rest/model/TableInfoModel.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/HColumnDescriptor.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/rest/model/CellSetModel.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestRowProcessorEndpoint.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/replication/regionserver/TestReplicationSink.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/MultipleColumnPrefixFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/HRegionInfo.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/rest/model/StorageClusterStatusModel.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/protobuf/TestProtobufUtil.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/InclusiveStopFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/SecureBulkLoadClient.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestBatchCoprocessorEndpoint.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestCoprocessorEndpoint.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/SingleColumnValueFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/BigDecimalColumnInterpreter.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/rest/model/ScannerModel.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/io/Reference.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/ColumnRangeFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/ipc/MasterCoprocessorRpcChannel.java * hbase-
protocol/src/main/java/com/google/protobuf/HBaseZeroCopyByteString.java * hbase-
client/src/test/java/org/apache/hadoop/hbase/client/TestClientNoCluster.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/ClusterStatus.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/DependentColumnFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPaginationFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/HTableDescriptor.java

107. SUCCESS: Integrated in HBase-0.98-on-Hadoop-1.1 #369 (See [https://builds.apache.org/job/HBase-0.98-on-Hadoop-1.1/369/]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString" (ndimiduk: rev 8884ad0444cada23dfbd4dbde9db9e2a20ae262a) *
hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPrefixFilter.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/regionserver/HRegionServer.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/ipc/MasterCoprocessorRpcChannel.java * hbase-
server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestCoprocessorEndpoint.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityController.java * hbase-
protocol/src/main/java/com/google/protobuf/HBaseZeroCopyByteString.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/filter/InclusiveStopFilter.java * hbase-
server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/HLogKey.java * hbase-
client/src/main/java/org/apache/hadoop/hbase/HColumnDescriptor.java * hbase-protocol/pom.xml * hbase-
client/src/main/java/org/apache/hadoop/hbase/protobuf/RequestConverter.java * hbase-

- client/src/main/java/org/apache/hadoop/hbase/filter/MultipleColumnPrefixFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/codec/MessageCodec.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/regionserver/TestPriorityRpc.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/FixedFileTrailer.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ByteArrayComparable.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFile.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/HTableDescriptor.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/zookeeper/ZKUtil.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/ScannerModel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ColumnPaginationFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/ipc/RegionCoprocessorRpcChannel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/HRegionInfo.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/CellModel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/ColumnRangeFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityUtils.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/FirstKeyValueMatchingQualifiersFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/SecureBulkLoadClient.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/SingleColumnValueFilter.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/hfile/HFileBlock.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/io/Reference.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/protobuf/ReplicationProtobufUtil.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestBatchCoprocessorEndpoint.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/regionserver/wal/SecureProtobufLogWriter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/ClusterStatus.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/protobuf/TestProtobufUtil.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/security/visibility/VisibilityClient.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/security/access/AccessControlClient.java * hbase-
 protocol/src/main/java/org/apache/hadoop/hbase/util/ByteString.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/TableInfoModel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/security/EncryptionUtil.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/DependentColumnFilter.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/filter/PrefixFilter.java * hbase-
 client/src/test/java/org/apache/hadoop/hbase/client/TestClientNoCluster.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/replication/regionserver/TestReplicationSink.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/StorageClusterStatusModel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/RegionTransition.java * hbase-
 server/src/test/java/org/apache/hadoop/hbase/coprocessor/TestRowProcessorEndpoint.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/rest/model/CellSetModel.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/client/coprocessor/BigDecimalColumnInterpreter.java
108. Still some 2,3 places in code, directly refer HBaseZeroCopyByteString rather than ByteString. I think all places we can change.
109. Addendum for master branch
110. +1 on addendum - some were due to recent checkins
111. Thanks Ted. master patch applies to branch-1 also. Attaching patch for 0.98 Will commit to all versions in some time unless objections.
112. Addendum pushed to all branches.
113. FAILURE: Integrated in HBase-1.0 #32 (See [<https://builds.apache.org/job/HBase-1.0/32/>]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString". -Addendum (anoopsamjohn: rev 15ba35b59330b408c76e18d9c8bb41329a11472b) * hbase-
 server/src/main/java/org/apache/hadoop/hbase/snapshot/SnapshotManifestV1.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/regionserver/RSRpcServices.java * hbase-
 server/src/main/java/org/apache/hadoop/hbase/snapshot/SnapshotManifestV2.java * hbase-
 client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java
114. Thanks Anoop.
115. FAILURE: Integrated in HBase-TRUNK #5293 (See [<https://builds.apache.org/job/HBase-TRUNK/5293/>]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString". -Addendum (anoopsamjohn: rev f5e13c7460dabd6909146e79f511af5bf7f5f977) * hbase-

- server/src/main/java/org/apache/hadoop/hbase/regionserver/RSRpcServices.java * hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java * hbase-server/src/main/java/org/apache/hadoop/hbase/snapshot/SnapshotManifestV2.java * hbase-server/src/main/java/org/apache/hadoop/hbase/snapshot/SnapshotManifestV1.java
116. FAILURE: Integrated in HBase-0.98 #390 (See [<https://builds.apache.org/job/HBase-0.98/390/>]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString". -Addendum (anoopsamjohn: rev cedd53a75eebe507da220785cb6c44e97300092c) * hbase-server/src/main/java/org/apache/hadoop/hbase/mapreduce/TableSnapshotInputFormatImpl.java * hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java
117. SUCCESS: Integrated in HBase-0.98-on-Hadoop-1.1 #370 (See [<https://builds.apache.org/job/HBase-0.98-on-Hadoop-1.1/370/>]) HBASE-11118 non environment variable solution for "IllegalAccessError: class com.google.protobuf.ZeroCopyLiteralByteString cannot access its superclass com.google.protobuf.LiteralByteString". -Addendum (anoopsamjohn: rev cedd53a75eebe507da220785cb6c44e97300092c) * hbase-client/src/main/java/org/apache/hadoop/hbase/protobuf/ProtobufUtil.java * hbase-server/src/main/java/org/apache/hadoop/hbase/mapreduce/TableSnapshotInputFormatImpl.java
118. FAILURE: Integrated in HBase-TRUNK #5304 (See [<https://builds.apache.org/job/HBase-TRUNK/5304/>]) Add note on HBASE-11118 and HBASE-10877 HBaseZeroCopyByteString woes to troubleshooting section (stack: rev bf2933e08ab65af3643bdadbc3d59c943d40602f) * src/main/docbkx/troubleshooting.xml
119. What is the workaround for running such application through Oozie? Setting HADOOP_CLASSPATH in Java and MapReduce actions are not possible. There seems to be no provision to do that.
120. hi [~ariforu]. Per the JIRA subject, after this commit, no environment variable manipulation is required. Are you seeing something different? Please send a note to the user@hbase list describing your hbase version, ooze, environment and any stack trace you're seeing in job launching. We'll help you get it resolved.
121. Thanks Nick for the quick response. We are using CDH5.1.0 with HBase 0.98.1. It seems from HBASE-11118 that this issue is fixed as of hbase 0.98.4 . Our application works fine when submitting directly from commandline after setting the following environment variable but it fails when spawned via Oozie. We have set the env variable in all data nodes, hadoop-env.sh for all users but without luck(probably being overridden somewhere). Wondering if there is any workaround for 0.98.1 with Oozie 4.0. export HADOOP_CLASSPATH=/usr/share/cmf/lib/cdh5/hbase-protocol-0.98.1-cdh5.1.0.jar:/etc/hbase/conf
122. Closing this issue after 0.99.0 release.