Item 13
git_comments:

1. * * Invoke 'getJar' on a JarFinder implementation. Useful for some job configuration * contexts (HBASE-8140) and also for testing on MRv2. First check if we have * HADOOP-9426. Lacking that, fall back to the backport. * * @param my_class the class to find. * @return a jar file that contains the class, or null.
2. function was properly called, but threw it's own exception. Unwrap it and pass it on.
3. * * Find a jar that contains a class of the same name, if any. * It will return a jar file, even if that is not the first thing * on the class path that has a class with the same name. * * This is shamelessly copied from JobConf * * @param my_class the class to find. * @return a jar file that contains the class, or null. * @throws IOException
4. attempt to locate an existing jar for the class.
5. URLDecoder is a misnamed class, since it actually decodes x-www-form-urlencoded MIME type rather than actual URL encoding (which the file path has). Therefore it would decode +s to ' 's which is incorrect (spaces are actually either unencoded or encoded as "%20"). Replace +s first, so that they are kept sacred during the decoding process.
6. toss all other exceptions, related to reflection failure

git_commits:

1. **summary:** HBASE-8158. Amend HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk)
   message: HBASE-8158. Amend HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk) git-svn-id: https://svn.apache.org/repos/asf/hbase/trunk@1465746 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Backport HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job"
   description: Assess the stability of the JarFinder util class from Hadoop-2, as a candidate for backporting to 0.94.

jira_issues_comments:

1. Attaching 0.94 patch from previous ticket.
2. Update patch to include addendum (adding SmallTests category).
3. Regarding backport risk (cc [~lhofhansl] [~stack]). This JarFinder utility has been in Hadoop for 18 months. Grep indicates is it used primarily by the MiniMRYarnCluster, which has seen a good deal of exercise in that time. Unfortunately, the only way to test this is via MR jobs run on distributed clusters, and our test automation doesn't cover this very well right now. I don't (yet) have a suite of MR jobs to act as a test battery. All this patch does is change the common code-patch to match the hadoop-0.23 codepath, as introduced in HBASE-5317. That patch shipped with 0.92.1. I think if there were issues we'd know about them by now.
4. Looks safe to backport IMO.
5. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12574608/8158-port-jarfinder-0.94.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 7 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/4919//console This message is automatically generated.
6. Maybe we can backport this, but still keep old findContainingJar(), and fall back to it. Smt like: - Try to locate JarFinder in the classpath, if found use it - Try backported JarFinder - If everything fails fallback to findContainingJar().
7. I see what you're saying, but your order of operations is off. I think you mean: - Try to locate JarFinder in the classpath, if found use it. (current behavior) - Try existing findContainingJar. (current behavior) - Fall back to backported JarFinder. (new behavior)
8. I though we might want to try the backported JarFinder first, but both should work. As long as the failure condition for both of them is "find the jar, throw exception, or return null" (vs finding a wrong jar), the ordering should not affect correctness.
9. JarFinder will build a jar from class files if a packages jar cannot be located. I believe the preference is to use an existing packaged jar over bundling one for the run. Hence, actually invoking it should be a last resort. This was *not* the behavior before I touched this bit of code ;)
10. bq. JarFinder will build a jar from class files if a packages jar cannot be located I see, lets do it in your order then.
11. Trunk should also be updated to match this behavior..?
12. bq. Trunk should also be updated to match this behavior..? I don't think so. If JarFinder is good at finding the jars, we should not need it. I proposed keeping it safe for 0.94.
13. The logic here is dumb! We should give priority to the common case, not the broken MRv2 test scenario. According to the existing comment around JarFinder, it was only needed on hadoop-0.23 for test cases. Indeed, JarFinder came out of hadoop-0.23 test code. Based on the that comment alone, it's not clear to me if MRv2 will ship the jar in the common case. I suggest the logic be: # try use existing {{findContainingJar}}. This will provide for normal MRv1 jobs (and maybe normal MRv2 jobs). # try use Backported {{JarFinder}}. This will provide for MRv1 "HCat Scenario" jobs, MRv2 test jobs (and maybe normal MRv2 jobs). # throw exception. It should fail configuring the job if the desired dependency cannot be found.
14. That is a shame. Hadoop should provide jar shipping utilities. Even JobConf.setJarByClass() uses findContainingJar(). Let's open a MAPREDUCE ticket, and get this util as a part of the public API. (independent of this ticket)
15. MAPREDUCE-5087 done.
16. This version applies the logic I described: # try using findContainingJar. # try backported JarFinder. # throw.
17. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12574675/8158-port-jarfinder-0.94.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 7 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/4927//console This message is automatically generated.
18. **body:** I have to admit that I am a bit confused about, I haven't been in this code a lot. If we find a jar finder class that ships with the version of hadoop that we are compiled against, shouldn't we use that?
    label: code-design
19. **body:** That would be ideal, yes. Trouble is, JarFinder is from hadoop-common-test, which means it's likely not on the classpath for prod deployments. The comment in that bit of code suggests it was originally introduced only for the benefit of MRv2 test cases. {noformat} $ find . -name JarFinder.java ./hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/JarFinder.java {noformat}
    label: code-design
20. MAPREDUCE-5087 recently became HADOOP-9426.
21. bump
22. What's it take to get a commit in these parts? FYI, assuming this version of the logic is accepted, I'll open a new ticket to bring 0.95/trunk to parity. (cc [~enis] [~lhofhansl] [~stack] [~jmspaggi])
23. In HADOOP-9426, we are trying to make JarFinder be a public main API. In this case, should not we still search for the JarFinder in the classpath, and only failover to backport as a last stand?
24. **body:** If we backport this in HBase, then why should we try to find it in Hadoop first? It will be a bit duplicate, no? It's the same code. Or do we have a longterm plan to remove this when all Hadoop versions will contain HADOOP-9426?
    label: code-design

25. [~enis] I'm a little concerned about feature drift between the forks, but the svn log indicates this class hasn't been touched much by Hadoop, so the likelihood is low. My preference is to explicitly use this version over using reflection to resolve one at run-time. [~jmspaggi] My preference is to delete any backports from HBase code once our support matrix of hadoop versions carries the necessary features.
26. I have an updated patch. Rebased, and resolving JarFinder via reflection per [~enis]'s comment. Testing it locally is a PITA because the integration test (HBASE-8147) isn't back-ported to 0.94. Will post shortly (I hope).
27. The logic seems good. However, in case we fall back to backport JarFinder, we shouldn't be accessing it through reflection. Can you just add the import and do a direct call.
28. This patch uses reflection only for hadoop's jarfinder.
29. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12576200/8158-port-jarfinder-0.94.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 7 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/5060//console This message is automatically generated.
30. **body:** This looks good to go, and Nick says that it fixes the IT test. The exact logic in the patch seems to be now: (1) Try existing findContainingJar. (2) Try to locate JarFinder in the classpath, if found use it. (3) Fall back to backported JarFinder. There is a reordering of steps 1 and 2 in the patch compared to current code, but it should not make a difference. Any objections? [~lhofhansl] do you want some time to take a look?
    label: code-design
31. Here's a patch for trunk that brings it into alignment with this new logic for 0.94.
32. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12576203/8158-trunk-parity.patch against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:red}-1 tests included{color}. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. {color:green}+1 hadoop2.0{color}. The patch compiles against the hadoop 2.0 profile. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {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 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:green}+1 core tests{color}. The patch passed unit tests in . Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-prefix-tree.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-client.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-common.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-protocol.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-server.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop1-compat.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-examples.html Findbugs warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//artifact/trunk/patchprocess/newPatchFindbugsWarningshbase-hadoop-compat.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/5062//console This message is automatically generated.
    label: code-design
33. +1 from me as well.

34. I have two +1's, yet [~lhofhansl] is conspicuously absent regarding the 0.94 backport. [~stack], [~apurtell], do you have an opinion for 0.96/trunk?
35. The trunk/0.95 patch is not factored for hadoop-compat and this seems like the sort of compatibility issue that was designed for. For 0.94 reflection makes sense. For 0.95+, the same code in hadoop-compat-1 makes sense, for hadoop-compat-2 we can assume JarFinder is there since Hadoop 2 code is >= 0.23. I would commit this to 0.94 (ping [~lhofhansl]) and maybe take it on the chin if there's test issues :-) because the patch looks good, the work on this issue looks good, and there has been good review. However I would like to hold off until a change goes in through trunk first. Please let me know your thoughts on the above comment.
36. Another way of stating the above is: Since this is a backport what is up with the trunk parity patch? Why is it not factored for hadoop-compat?
37. Further development when into this patch than in the initial version on trunk. Notably, the precise order of class look-up for jar discovery is eventually different on this ticket than what went into trunk, hence the parity patch. Trouble is, JarFinder isn't actually provided by released hadoop yet, on any version. It might be on the classpath in a test jar, but not in a production deployment. IMHO, the reflection code should not have been added as it was, because it was a work-around for a broken test environment. We now have a legitimate use-case for this same code-path, hence it gaining some new attention. I'll take a look at the hadoop-compat modules on 0.95/trunk and see if I can make sense of a variation, update the parity patch accordingly. The 0.94 patch still applies. We'll still need to ship the backported JarFinder for 1.x hadoops and, unless HADOOP-9426 squeezes in soon, for early releases of the 2.x line as well.
38. bq. We'll still need to ship the backported JarFinder for 1.x hadoops and, unless HADOOP-9426 squeezes in soon, for early releases of the 2.x line as well. Thanks for the pointer to HADOOP-9426, I'll commit the patches on this issue to trunk, 0.95, and 0.94 tomorrow unless objection.
39. Looks right to me (but I cannot possible foresee all the scenarios we'll run M/R). With caveat... +1
40. [~lhofhansl] There have been a couple of commits to 0.94 since 0.94.6. Presume you'll be branching 0.94.6.1 from the 0.94.6 release tag. If so, committing this to 0.94 branch won't be an issue there. Please advise.
41. Bump.
42. Looks safe to commit now.
43. Committed to trunk, 0.95, and 0.94. TestTableMapReduce and TestJarFinder pass locally when applied.
44. Thanks [~apurtell]!
45. Integrated in HBase-0.94 #951 (See [https://builds.apache.org/job/HBase-0.94/951/]) HBASE-8158. Backport HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk) (Revision 1465743) Result = SUCCESS apurtell : Files : * /hbase/branches/0.94/src/main/java/org/apache/hadoop/hbase/mapreduce/TableMapReduceUtil.java * /hbase/branches/0.94/src/main/java/org/apache/hadoop/hbase/mapreduce/hadoopbackport/JarFinder.java * /hbase/branches/0.94/src/test/java/org/apache/hadoop/hbase/mapreduce/hadoopbackport * /hbase/branches/0.94/src/test/java/org/apache/hadoop/hbase/mapreduce/hadoopbackport/TestJarFinder.java
46. Integrated in HBase-TRUNK #4044 (See [https://builds.apache.org/job/HBase-TRUNK/4044/]) HBASE-8158. Amend HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk) (Revision 1465746) Result = FAILURE apurtell : Files : * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/mapreduce/TableMapReduceUtil.java
47. Integrated in hbase-0.95-on-hadoop2 #61 (See [https://builds.apache.org/job/hbase-0.95-on-hadoop2/61/]) HBASE-8158. Amend HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk) (Revision 1465745) Result = FAILURE apurtell : Files : * /hbase/branches/0.95/hbase-server/src/main/java/org/apache/hadoop/hbase/mapreduce/TableMapReduceUtil.java
48. Integrated in HBase-TRUNK-on-Hadoop-2.0.0 #488 (See [https://builds.apache.org/job/HBase-TRUNK-on-Hadoop-2.0.0/488/]) HBASE-8158. Amend HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk) (Revision 1465746) Result = FAILURE apurtell : Files : * /hbase/trunk/hbase-server/src/main/java/org/apache/hadoop/hbase/mapreduce/TableMapReduceUtil.java
49. Integrated in HBase-0.94-security #133 (See [https://builds.apache.org/job/HBase-0.94-security/133/]) HBASE-8158. Backport HBASE-8140 "TableMapReduceUtils#addDependencyJar fails when nested inside another MR job" (Nick Dimiduk) (Revision 1465743) Result = SUCCESS apurtell : Files : * /hbase/branches/0.94/src/main/java/org/apache/hadoop/hbase/mapreduce/TableMapReduceUtil.java * /hbase/branches/0.94/src/main/java/org/apache/hadoop/hbase/mapreduce/hadoopbackport/JarFinder.java * /hbase/branches/0.94/src/test/java/org/apache/hadoop/hbase/mapreduce/hadoopbackport * /hbase/branches/0.94/src/test/java/org/apache/hadoop/hbase/mapreduce/hadoopbackport/TestJarFinder.java