Item 139
git_comments:

1. use org.openlabtesting.leveldbjni on aarch64 platform
2. org.fusesource.leveldbjni will be used except on arm64 platform.

git_commits:

1. **summary:** [SPARK-27721][BUILD] Switch to use right leveldbjni according to the platforms
   message: [SPARK-27721][BUILD] Switch to use right leveldbjni according to the platforms This change adds a profile to switch to use the right leveldbjni package according to the platforms: aarch64 uses org.openlabtesting.leveldbjni:leveldbjni-all.1.8, and other platforms use the old one org.fusesource.leveldbjni:leveldbjni-all.1.8. And because some hadoop dependencies packages are also depend on org.fusesource.leveldbjni:leveldbjni-all, but hadoop merge the similar change on trunk, details see https://issues.apache.org/jira/browse/HADOOP-16614, so exclude the dependency of org.fusesource.leveldbjni for these hadoop packages related. Then Spark can build/test on aarch64 platform successfully. Closes #26636 from huangtianhua/add-aarch64-leveldbjni. Authored-by: huangtianhua <huangtianhua@huawei.com> Signed-off-by: Sean Owen <sean.owen@databricks.com>

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-27721][BUILD] Switch to use right leveldbjni according to the platforms
   body: This change adds a profile to switch to use the right leveldbjni package according to the platforms: aarch64 uses org.openlabtesting.leveldbjni:leveldbjni-all.1.8, and other platforms use the old one org.fusesource.leveldbjni:leveldbjni-all.1.8. And because some hadoop dependencies packages are also depend on org.fusesource.leveldbjni:leveldbjni-all, but hadoop merge the similar change on trunk, details see https://issues.apache.org/jira/browse/HADOOP-16614, so exclude the dependency of org.fusesource.leveldbjni for these hadoop packages related. Then Spark can build/test on aarch64 platform successfully.

github_pulls_comments:

1. **[Test build #4942 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/NewSparkPullRequestBuilder/4942/testReport)** for PR 26636 at commit [`f249e38`](https://github.com/apache/spark/commit/f249e3890196e1d2dac104f0ffa2e467546e5152). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
2. **[Test build #4952 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/NewSparkPullRequestBuilder/4952/testReport)** for PR 26636 at commit [`65843b8`](https://github.com/apache/spark/commit/65843b8a2856aa56051209ece0abc33da77ca0ab). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
3. Merged to master

github_pulls_reviews:

1. I'm not sure we want this; this makes the build depend on the machine that built it. It really depends on the target architecture. You'd activate this with `-Paarch64` then, from any build machine.
2. Not a big deal, but I'd put this farther up where versions are managed, at least.
3. Fisrt, unforturnately, the main repo seems is not under maintaining, I've asked committers to public a new release which supports aarch64 platform, but there is no reply. Second, if the main repo have 1.9 version, I believe the 1.9 version should support aarch64 platform, then I think we can migrate to use it directly? And last, if you want to use org.openlabtesting.leveldbjni:leveldbjni-all.1.9 I will make it :) About the 'activation' I think it only specify the default profile depends on the build machine, you can activete with -Paarch64 on any other build machine.
4. You mean to add another property like 'leveldbjni.version'?
5. Actually, you've had prompt responses repeatedly: https://mail-archives.apache.org/mod_mbox/spark-dev/201911.mbox/%3CCAEccTyxQ31BVP3vty0h_VxE_%3DfE7OWUg%3DqpumuZuK5kDys21JQ%40mail.gmail.com%3E I won't reply a fourth time, but here, you're suggesting there is still something that needs to be done to make aarch64 work. Why would anyone announce it works then? No, the project is not going to take on this goal for you; you are best placed to do this work. As before, you are getting prompt attention for your efforts. I do not expect that Spark will ever release a separate aarch64 build. So, the ideal solution is to make the upstream version compatible with all architectures. I do not know what the main leveldbjni project is or isn't going to support; it's what I'm asking you. For now, relying on a fork under a profile could be OK. Are there any other differences in the fork? Yes, you do not want to activate based on the build machine's architecture. It should be specified manually.
6. No. I mean move this up to where versions are managed. It doesn't make as much sense to put it at the bottom with this test and JVM properties.
7. @srowen, I am sorry didn't say it clear, 'I've asked committers to public a new release which supports aarch64 platform, but there is no reply.' ---- I said I've ask the org.fusesource.leveldbjni to public a new release(like 1.9) to support aarch64, not spark:)

8. @srowen, and about the spark arm CI, now I will follow your suggestion only to send an email to user@ tell users the situation of the arm testing of spark.
9. Oh I entirely misread that. Please disregard the first comment with my apologies.
10. Are there any other differences in the fork? --- @srowen, I fork fusesource/leveldbjni to theopenlab/leveldbjni, and yes, we've modified it to make it support aarch64, the details you can see https://github.com/theopenlab/leveldbjni/commit/aa6b14a50a9acb2681102975107784f052f1ceeb (the code refers to google/leveldb https://github.com/google/leveldb/commit/c4c38f9c1f3bb405fe22a79c5611438f91208d09#diff-b20d9097440605779c4984e3a1346085) I only built new .so for aarch64 platform.
11. fusesource/leveldbjni:leveldbjni-all.1.8 supports platforms: osx, linux32, linux64(x86), win32,win64, see https://mvnrepository.com/artifact/org.fusesource.leveldbjni/leveldbjni-all/1.8 we add linux64-aarch64 to openlabtesting/leveldbjni:leveldbjni-all.1.8, see https://mvnrepository.com/artifact/org.openlabtesting.leveldbjni/leveldbjni-all/1.8
12. @srowen, so is there possible migrate to use openlabtesting.leveldbjni:leveldbjni-all.1.8 directly in spark for all platforms supported?
13. ok, will move it up.
14. The most ideal outcome would be for the upstream project to adopt the change. It looks quite minor. Otherwise the fork might end up running behind upstream changes, but it sounds like there is not much upstream activity. Next-most ideal would be for Hadoop to adopt the same approach and we use a release that follows the same pattern. But the approach here is OK and maybe less intrusive as it won't affect the main build.
15. https://issues.apache.org/jira/browse/HADOOP-16614 hadoop adopt the same approach, the code has been merged.
16. I know it's minor, but, I would also not stick it in the middle of the Hive-related settings. Just maybe put it after arrow.version or something.
17. ok, thanks, will do it.

jira_issues:

1. **summary:** Missing leveldbjni package of aarch64 platform
   description: Currently, Hadoop denpend on the *leveldbjni-all:1.8* package of *org.fusesource.leveldbjni* group, but it cannot support ARM platform. see: [https://search.maven.org/search?q=g:org.fusesource.leveldbjni] Because the leveldbjni community is inactivity and the code ([https://github.com/fusesource/leveldbjni]) didn't updated a long time.I will build the leveldbjni package of aarch64 platform, and upload it with other platform packages of *org.fusesource.leveldbjni* to a new *org.openlabtesting.leveldbjni* maven repo. In hadoop code, I will add a new profile aarch64 for for automatically select the *org.openlabtesting.leveldbjni* artifact group and using the aarch64 package of leveldbjni when running on ARM server, this approach has no effect on current code.

jira_issues_comments:

1. The patch looks good to me. I will commit if no objections.
2. Thank you [~seanlau] for the patch. +1 merged to trunk.
3. SUCCESS: Integrated in Jenkins build Hadoop-trunk-Commit #17568 (See [https://builds.apache.org/job/Hadoop-trunk-Commit/17568/]) HADOOP-16614. Add aarch64 support for dependent leveldbjni. (eyang: rev ac6b6a6a85c126efbfda12dc9979706490246bbe) * (edit) hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-shuffle/pom.xml * (edit) hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/pom.xml * (edit) hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-hs/pom.xml * (edit) hadoop-mapreduce-project/pom.xml * (edit) hadoop-project/pom.xml * (edit) hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/pom.xml * (edit) hadoop-client-modules/hadoop-client-minicluster/pom.xml * (edit) pom.xml * (edit) hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-applicationhistoryservice/pom.xml * (edit) hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/pom.xml * (edit) hadoop-hdfs-project/hadoop-hdfs/pom.xml
4. Do we plan to backport this to 2.7 and 3.2? Spark has the same issue on arm platform, it depends on *org.fusesource.leveldbjni* not only itself, but also it depends on some hadoop 2.7(or 3.2) jar packages which using *org.fusesource.leveldbjni*, so it will be good that we backport this to 2.7 and 3.2 :)