Item 23

git_comments:

1. * * @return true when running JVM is having sun's Unsafe package available in it and underlying * system having unaligned-access capability.
2. * * @return true when running JVM is having sun's Unsafe package available in it and underlying * system having unaligned-access capability.
3. Using java.nio.Bits#unaligned() to check for unaligned-access capability

git_commits:

1. **summary:** HBASE-14940 Make our unsafe based ops more safe.
   message: HBASE-14940 Make our unsafe based ops more safe.

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Make our unsafe based ops more safe
   description: Thanks for the nice findings [~ikeda] This jira solves 3 issues with Unsafe operations and ByteBufferUtils 1. We can do sun unsafe based reads and writes iff unsafe package is available and underlying platform is having unaligned-access capability. But we were missing the second check 2. Java NIO is doing a chunk based copy while doing Unsafe copyMemory. The max chunk size is 1 MB. This is done for "A limit is imposed to allow for safepoint polling during a large copy" as mentioned in comments in Bits.java. We are also going to do same way 3. In ByteBufferUtils, when Unsafe is not available and ByteBuffers are off heap, we were doing byte by byte operation (read/copy). We can avoid this and do better way.

jira_issues_comments:

1. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12776082/HBASE-14940.patch against master branch at commit 8bf70144e40650ef972f005e2465bd0e2a087c40. ATTACHMENT ID: 12776082 {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 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.6.1 2.7.0 2.7.1) {color:red}-1 javac{color}. The applied patch generated 36 javac compiler warnings (more than the master's current 35 warnings). {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not generate new checkstyle errors. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) 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 post-site goal succeeds with this patch. {color:green}+1 core tests{color}. The patch passed unit tests in . {color:green}+1 zombies{color}. No zombie tests found running at the end of the build. Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/16782//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/16782//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/16782//artifact/patchprocess/checkstyle-

aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/16782//console This message is automatically generated.
label: code-design

2. **body:** If there is not unaligned capability, when the starting point to read/write is at an appropriate aligned boundary it seems possible to use Unsafe methods. It might be tedious to write such code for every method and its effect might be doubtful, but in the bulk comparison inserting fine reading until reaching an appropriate aligned boundary might be worth. In the Oracle implementation, methods to read/write a single data just use the condition "unaligned", but DirectDirectBuffer.asInt/LongBuffer etc. methods also check the boundary in order to, I think, prepare effective bulk read/write. Comparison to true/false in {{if}} conditions is confusing. It is appropriate to use the operator "!" instead of comparison to false. FusszyRowFilter might be appropriate to using polymorphism, making it an abstract class with a static factory method.
   label: code-design
3. **body:** {quote} Comparison to true/false in {{if}} conditions is confusing. It is appropriate to use the operator "!" instead of comparison to false. FusszyRowFilter might be appropriate to using polymorphism, making it an abstract class with a static factory method. {quote} All existing code.. Only change is checking whether unaligned rather than Unsafe avail.. So FusszyRowFilter changes we can do in another IA jira.. You mind doing such cleanup? If so pls raise a Jira.
   label: code-design
4. **body:** bq, if (UnsafeAccess.unaligned() == false) { The above readst wrong but my guess is that the author was trying to make a point given the odd name of the method. In unsafeCopy if len > UNSAFE_COPY_THRESHOLD we will call it again to finish up the copy? [~ikeda] We don't want the range checking because we think we know where we are (famous last words) and it costs. Hopefully we won't be spending too much time debugging corrupt memory reads...
   label: code-design
5. bq.In unsafeCopy if len > UNSAFE_COPY_THRESHOLD we will call it again to finish up the copy? Yes. It will copy in chunks of max 1 MB.. Just as in java NIO package..
6. I'm +1 on the patch. What you think [~ikeda]? Commit this and then file new issues to probe around reads when no aligned facility and/or effects of not doing boundary checks?
7. There are risks (from the beginning) and if I were I might rewrite whole classes. As for aligned boundary checks I have second thoughts that is overdoing and it is enough to just call the corresponding API for fallback. I rather prefer to respect the decision by VM developers.
8. **body:** Regarding the COPY_THRESHOLD thing that VM does, I could see many other projects that deal with Unsafe also omits that check. Including the alignment but handles the Endian part. Just saying this for info. So may be other code bases can also see how this can be done. But from me I feel this is ok as the alignment is handled here and if there is no alignment then we fall back to the normal way right?
   label: code-design
9. bq.There are risks (from the beginning) We are adding the missing checks which NIO was doing to make use of the Unsafe. So what else you see? I think we are better with this and plan to commit tonight IST. Any other improvements, we can raise follow up jiras.
10. **body:** {color:red}-1 overall{color}. Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/16922//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/16922//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/16922//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/16922//console This message is automatically generated.
    label: code-design
11. Is this shortened format your doing [~busbey] ? If not, I'll dig in. Let me rerun patch...
12. Retry I think the curtailed format is my fault. Let me fix.
13. the short format isn't from me, but that's how it's been showing up when the buildbot runs fall over.
14. I filed HBASE-15009. Some commit to test-patch on branch broke the build up of the JIRA_COMMENT variable content. I was last commit. One before mine, messed with JIRA_COMMENT. Let me just try and get it same as master branch and then you can fix it proper w/ yetus [~busbey]
15. **body:** {color:red}-1 overall{color}. Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/16924//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/16924//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/16924//artifact/patchprocess/checkstyle-

aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/16924//console This message is automatically generated.
label: code-design

16. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12778642/HBASE-14940_branch-1.patch against branch-1 branch at commit cdca22a36d0e2ec87cea732e0769c4fa4fa37e57. ATTACHMENT ID: 12778642 {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 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.6.1 2.7.0 2.7.1) {color:red}-1 javac{color}. The applied patch generated 17 javac compiler warnings (more than the master's current 15 warnings). {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not generate new checkstyle errors. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) 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 post-site goal succeeds with this patch. {color:green}+1 core tests{color}. The patch passed unit tests in . {color:green}+1 zombies{color}. No zombie tests found running at the end of the build. Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/16936//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/16936//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/16936//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/16936//console This message is automatically generated.
    label: code-design
17. Same branch-1 patch applies to 0.98+ versions.. I think we need this in 0.98+ versions. Any objections?
18. Not from me.
19. **body:** The code for checking unalignedness was copied from Java. This part should be rewritten using reflection due to copyright.
    label: code-design
20. bq. This part should be rewritten using reflection due to copyright. What part? Marking unaligned dependent on arch? How would you do that one-liner with reflection? If you search this line on the web, there is at least one instance where this line has a compatible license....
21. The required code is in Bits.java#unaligned() which is a static private method. Can do reflection call on that. As it is a private class and private method in Java in future if Java make any changes to Class name or method name or signature, we may land in issues.. That is why I decided to copy paste the code. So if there is a consensus to use Reflection based look up, I can change it. Patch is committed to trunk already. I can do the needful change as addendum commit.
22. That is a license issue, not a compatibility issue. The comment {quote} + // Copied from java.nio.Bits.java#unaligned {quote} explicit says that, though we could say a quibble that we don't specify which Java implementation we copied the code from :P Copying from OpenJDK causes GPL pollution, if I'm not mistaken. I'm not sure about Oracle JDK.
23. +1 on v2
24. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12779175/HBASE-14940_v2.patch against master branch at commit 1af98f255132ef6716a1f6ba1d8d71a36ea38840. ATTACHMENT ID: 12779175 {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 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.6.1 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not generate new checkstyle errors. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) 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 post-site goal succeeds with this patch. {color:green}+1 core tests{color}. The patch passed unit tests in . {color:green}+1 zombies{color}. No zombie tests found running at the end of the build. Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/16987//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/16987//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/16987//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/16987//console This message is automatically generated.
label: code-design

25. **body:** {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12779274/HBASE-14940_branch-1.patch against branch-1 branch at commit e00a04df10de70b029a2d1f115f97f9d79a05c6a. ATTACHMENT ID: 12779274 {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 hadoop versions{color}. The patch compiles with all supported hadoop versions (2.4.0 2.4.1 2.5.0 2.5.1 2.5.2 2.6.0 2.6.1 2.7.0 2.7.1) {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 protoc{color}. The applied patch does not increase the total number of protoc compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 checkstyle{color}. The applied patch does not generate new checkstyle errors. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) 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 post-site goal succeeds with this patch. {color:red}-1 core tests{color}. The patch failed these unit tests: org.apache.hadoop.hbase.security.token.TestGenerateDelegationToken {color:green}+1 zombies{color}. No zombie tests found running at the end of the build. Test results: https://builds.apache.org/job/PreCommit-HBASE-Build/16994//testReport/ Release Findbugs (version 2.0.3) warnings: https://builds.apache.org/job/PreCommit-HBASE-Build/16994//artifact/patchprocess/newFindbugsWarnings.html Checkstyle Errors: https://builds.apache.org/job/PreCommit-HBASE-Build/16994//artifact/patchprocess/checkstyle-aggregate.html Console output: https://builds.apache.org/job/PreCommit-HBASE-Build/16994//console This message is automatically generated.
    label: code-design
26. bq.org.apache.hadoop.hbase.security.token.TestGenerateDelegationToken Test not related to this patch. Will commit to 0.98+ branches.
27. Pushed to 0.98+ versions. Thanks all for the reviews
28. SUCCESS: Integrated in HBase-1.3-IT #402 (See [https://builds.apache.org/job/HBase-1.3-IT/402/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 4a7565af9cf8ef7e40ef3c592d6815d1b671fb5e) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java
29. SUCCESS: Integrated in HBase-1.2-IT #362 (See [https://builds.apache.org/job/HBase-1.2-IT/362/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 9b459cddb791832873cfc9f84e6dcfc0484be617) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java
30. SUCCESS: Integrated in HBase-1.2 #470 (See [https://builds.apache.org/job/HBase-1.2/470/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 9b459cddb791832873cfc9f84e6dcfc0484be617) * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java

31. FAILURE: Integrated in HBase-1.3 #466 (See [https://builds.apache.org/job/HBase-1.3/466/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 4a7565af9cf8ef7e40ef3c592d6815d1b671fb5e) * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java
32. FAILURE: Integrated in HBase-1.1-JDK8 #1715 (See [https://builds.apache.org/job/HBase-1.1-JDK8/1715/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 6b5645e37a96109c155c681b330132c74c0bd7b6) * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java
33. FAILURE: Integrated in HBase-1.1-JDK7 #1628 (See [https://builds.apache.org/job/HBase-1.1-JDK7/1628/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 6b5645e37a96109c155c681b330132c74c0bd7b6) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java
34. FAILURE: Integrated in HBase-Trunk_matrix #583 (See [https://builds.apache.org/job/HBase-Trunk_matrix/583/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 6fc2596ab37614fe35ccfebda0564e4721bd4b95) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/ByteBufferUtils.java * hbase-common/src/main/java/org/apache/hadoop/hbase/nio/SingleByteBuff.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java
35. FAILURE: Integrated in HBase-0.98-on-Hadoop-1.1 #1150 (See [https://builds.apache.org/job/HBase-0.98-on-Hadoop-1.1/1150/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev f39c41ffe56ae6dab80397651eb97fc5de8a9ab3) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java
36. FAILURE: Integrated in HBase-0.98-matrix #277 (See [https://builds.apache.org/job/HBase-0.98-matrix/277/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev f39c41ffe56ae6dab80397651eb97fc5de8a9ab3) * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java
37. It looks like below only called on instantiation of UnsafeAccess... {code} if (theUnsafe != null) { 64 BYTE_ARRAY_BASE_OFFSET = theUnsafe.arrayBaseOffset(byte[].class); 71 BYTE_ARRAY_BASE_OFFSET = theUnsafe.arrayBaseOffset(byte[].class); 72 try { 73 // Using java.nio.Bits#unaligned() to check for unaligned-access capability 74 Class<?> clazz = Class.forName("java.nio.Bits"); 75 Method m = clazz.getDeclaredMethod("unaligned"); 76 m.setAccessible(true); 77 unaligned = (boolean) m.invoke(null); 78 } catch (Exception e) { 79 unaligned = false; 80 } {code} i.e. we do the reflection once only? Patch seems good otherwise.
38. FAILURE: Integrated in HBase-1.0 #1127 (See [https://builds.apache.org/job/HBase-1.0/1127/]) HBASE-14940 Make our unsafe based ops more safe. (anoopsamjohn: rev 86d4e2084d354f12d21f993464f88ec286a8a594) * hbase-client/src/main/java/org/apache/hadoop/hbase/filter/FuzzyRowFilter.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java
39. Looks like this break compilation in 0.98 now: {code} [ERROR] /home/lars/dev/hbase-0.98/hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java:[67,38] error: incompatible types: Object cannot be converted to boolean {code}
40. Must have to do with compileSource being 1.6 in 0.98, and 1.7+ everywhere else.
41. Should we replace {source} unaligned = (boolean) m.invoke(null); {source} with {source} unaligned = (Boolean) m.invoke(null); {source} In 0.98?
42. Yes only at the load time of UnsafeAccess, we do make unsafe ref (if sun.Unsafe available) and then check for unalign capability. This is same way as in java nio.

43. Sound good [~lhofhansl] if it works sir
44. Sorry for the trouble.. Reopen issue for the 0.98 addendum
45. As per Lars suggestion.
46. +1 on addendum. I'm going to commit that to 0.98. (It works for me compiling with JDK 1.8.)
47. TestFromClientSide passes. If anything was wrong this, I assume that would fail.
48. And looks right in the debugger.
49. FAILURE: Integrated in HBase-0.98-matrix #278 (See [https://builds.apache.org/job/HBase-0.98-matrix/278/]) HBASE-14940 addendum for 0.98. (larsh: rev dbd8c94536553570e0febc77e69aa7cd93d58f1c) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java
50. FAILURE: Integrated in HBase-0.98-on-Hadoop-1.1 #1151 (See [https://builds.apache.org/job/HBase-0.98-on-Hadoop-1.1/1151/]) HBASE-14940 addendum for 0.98. (larsh: rev dbd8c94536553570e0febc77e69aa7cd93d58f1c) * hbase-common/src/main/java/org/apache/hadoop/hbase/util/UnsafeAccess.java
51. Bulk closing 1.1.3 issues.