Item 379
git_comments:

git_commits:

1. **summary:** ORC-334: [C++] Add AppVeyor support for integration on windows
   message: ORC-334: [C++] Add AppVeyor support for integration on windows Fixes #265 Signed-off-by: Deepak Majeti <mdeepak@apache.org>

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
2. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
3. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
4. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
5. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
6. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
7. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
8. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
9. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
   body:
10. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
    label: code-design
11. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
    label: code-design
12. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
13. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
14. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
15. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
16. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
17. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:
18. **title:** ORC-334: [C++] Add AppVeyor support for integration on windows
    body:

github_pulls_comments:

1. AppVeyor can be enabled after this PR: https://ci.appveyor.com/project/rip-nsk/orc/build/1.0.33 Notes (for windows env): - Library depends on tzdata package (taken from cygwin.osuosl.org). - LocalTimezone is hardcoded to "UTC" -

CompressionKind_MAX and WriterVersion_MAX are changed to INT32_MAX (enum is int in MSVC) - Probably timegm, gmtime_r and asctime_r should be moved to Adaptor.cc, or implemented as inline functions - BUILD_TOOLS=OFF

2. +1 LGTM
3. It seems AppVeyor should be enabled by apache/orc project owner here - https://ci.appveyor.com/projects.
4. Added an INFRA ticket to do this here https://issues.apache.org/jira/browse/INFRA-16535

github_pulls_reviews:

1. We generally prefer to use explicitly sized types, so uint64_t or uint32_t would be better.
2. Can you comment on why we look at `UTC` for windows instead of `LOCAL_TIMEZONE`?
3. Because there isn't a direct windows equivalent. Here is what Java writes about it- https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/time-zone002.html There is a registry setting, but it uses a non-standard set of names *doh*. The easiest way would be to create a Java jar and run that to determine the local timezone.
4. http://unicode.org/cldr/data/common/supplemental/windowsZones.xml - here is a mapping of windows <=> *nix time zones, and here - https://github.com/google/cctz - "timezone" library which probably is a best choice (BTW https://github.com/google/cctz/issues/53). Anyway, it is non trivial work which I think does not related to orc directly. I believe it should be customer decision how to pass/extract timestamps from orc library and any timezone conversions should be done on customer side. Hardcoding local timezone to UTC just unblock windows build (w/o breaking current API in general).
5. **body:** Recently we have change to UTC everywhere in C++ code. Is it better to delete getLocalTimezone function to avoid confusion?
   label: code-design
6. **body:** getLocalTimezone is still used in RowReaderImpl. Probably it should be replaced with getDefaultTimezone() { return getTimezoneByName("UTC"); }
   label: code-design
7. Just double checked, we should still use local timezone if writer timezone is not found to keep backward compatibility.
8. Why it (local timezone) can't be UTC in this (or any other) case?
9. If we cannot find writer timezone from the stripe footer, we assume the writer and reader are in the same timezone. TimestampColumnReader tries to convert the timestamp to UTC to get the same wall clock time as the writer. Unless the files are written under UTC, otherwise we may get wrong timestamps if we set local timezone to UTC by default.
10. So, proposed solution (local == UTC) will work until we read of the file created using java writer.
11. Any objections on future use cctz library?
12. I also think that it is better to leave the conversion to the client/customer. We should ideally change the conversion for Nix* systems but cannot due to backward compatibility. We should be able to converge both Java and C++ timestamps in ORC 2.0. For Windows, since this is the first official build support, we should be okay to use "UTC" and document this behavior. I will merge this PR end of today if there are no objections.

jira_issues:

1. **summary:** Add AppVeyor support for integration on windows
   description: Like Travis CI, we must integrate AppVeyor for windows testing.
2. **summary:** Add AppVeyor support for integration on windows
   description: Like Travis CI, we must integrate AppVeyor for windows testing.
3. **summary:** Add AppVeyor support for integration on windows
   description: Like Travis CI, we must integrate AppVeyor for windows testing.
4. **summary:** Add AppVeyor support for integration on windows
   description: Like Travis CI, we must integrate AppVeyor for windows testing.

jira_issues_comments:

1. I have a draft for this in my fork [https://github.com/rip-nsk/orc/tree/ORC-TBD___C%2B%2B%5D_appveyor|http://example.com] [https://ci.appveyor.com/project/rip-nsk/orc/build/1.0.16|http://example.com]

2. GitHub user rip-nsk opened a pull request: https://github.com/apache/orc/pull/265 ORC-334: [C++] Add AppVeyor support for integration on windows You can merge this pull request into a Git repository by running: $ git pull https://github.com/rip-nsk/orc ORC-334 Alternatively you can review and apply these changes as the patch at: https://github.com/apache/orc/pull/265.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #265 ---- commit 916d5eece4c3483b4a5a639d81b0536bd88aaa41 Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:15:40Z s/uint/unsigned/ commit 846cb2ebda368e2b5d1efccf2f30b8d13d151c99 Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:16:37Z s/bzero/memset commit 3d44b6a6bdad89897ded341a7c9913582716536c Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:21:22Z IANA - Time Zone Database commit 0e16111b00c9bb9dc5f27980fc95dd7d49ac76a0 Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:21:49Z Adaptor.hh patch commit 2fb7b3457612418b22f4e15ffa8eda7aa21cdecc Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:22:29Z Timezone.cc patch commit a856d82be5b39c6eea84a9c78f7ecf8e5619d574 Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:23:08Z Common.hh patch commit c35e06e0b27e0a7a110d474aa1c1ba6c19e0e316 Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:23:42Z appveyor.yml commit 6db24812e3032a9f551c633e87e096cc2d6efc5f Author: rip-nsk <rip.nsk@...> Date: 2018-05-03T23:47:45Z CMAKE_BUILD_TYPE ----
3. Github user rip-nsk commented on the issue: https://github.com/apache/orc/pull/265 AppVeyor after this PR: [https://ci.appveyor.com/project/rip-nsk/orc/build/1.0.32]
4. Github user omalley commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186114158 --- Diff: c++/test/CreateTestFiles.cc --- @@ -31,8 +31,8 @@ void writeCustomOrcFile(const std::string& filename, const orc::proto::Metadata& metadata, const orc::proto::Footer& footer, - const std::vector<uint>& version, - uint writerVersion) { + const std::vector<unsigned>& version, --- End diff -- We generally prefer to use explicitly sized types, so uint64_t or uint32_t would be better.
5. Github user majetideepak commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186409327 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- Can you comment on why we look at `UTC` for windows instead of `LOCAL_TIMEZONE`?
6. Github user omalley commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186474701 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- Because there isn't a direct windows equivalent. Here is what Java writes about it- https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/time-zone002.html There is a registry setting, but it uses a non-standard set of names *doh*. The easiest way would be to create a Java jar and run that to determine the local timezone.
7. Github user rip-nsk commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186487106 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- http://unicode.org/cldr/data/common/supplemental/windowsZones.xml - here is a mapping of windows <=> *nix time zones, and here - https://github.com/google/cctz - "timezone" library which probably is a best choice (BTW https://github.com/google/cctz/issues/53). Anyway, it is non trivial work which I think does not related to orc directly. I believe it should be customer decision how to pass/extract timestamps from orc library and any timezone conversions should be done on customer side.
8. Github user wgtmac commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186492444 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- Recently we have change to UTC everywhere in C++ code. Is it better to delete getLocalTimezone function to avoid confusion?
9. Github user rip-nsk commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186493219 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- getLocalTimezone is still used in RowReaderImpl
10. Github user wgtmac commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186827146 --- Diff: c++/src/Timezone.cc --- @@

-710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- Just double checked, we should still use local timezone if writer timezone is not found to keep backward compatibility.

11. Github user rip-nsk commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186828362 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- Why it can't be UTC in this (or any other) case?
12. Github user wgtmac commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186840580 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- If we cannot find writer timezone from the stripe footer, we assume the writer and reader are in the same timezone. TimestampColumnReader tries to convert the timestamp to UTC to get the same wall clock time as the writer. Unless the files are written under UTC, otherwise we may get wrong timestamps if we set local timezone to UTC by default.
13. Github user rip-nsk commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r186869413 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- So, proposed solution (local == UTC) will work until we read of the file created using java writer.
14. Github user rip-nsk commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r187164583 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- Any concern on future use cctz library?
15. Github user majetideepak commented on a diff in the pull request: https://github.com/apache/orc/pull/265#discussion_r187922554 --- Diff: c++/src/Timezone.cc --- @@ -710,7 +710,11 @@ namespace orc { * Get the local timezone. */ const Timezone& getLocalTimezone() { +#ifdef _MSC_VER + return getTimezoneByName("UTC"); --- End diff -- I also think that it is better to leave the conversion to the client/customer. We should ideally change the conversion for Nix* systems but cannot due to backward compatibility. We should be able to converge both Java and C++ timestamps in ORC 2.0. For Windows, since this is the first official build support, we should be okay to use "UTC" and document this behavior. I will merge this PR end of today if there are no objections.
16. Github user majetideepak commented on the issue: https://github.com/apache/orc/pull/265 +1 LGTM
17. Github user asfgit closed the pull request at: https://github.com/apache/orc/pull/265
18. Github user rip-nsk commented on the issue: https://github.com/apache/orc/pull/265 It seems AppVeyor should be enabled by apache/orc project owner here - https://ci.appveyor.com/projects.
19. Github user majetideepak commented on the issue: https://github.com/apache/orc/pull/265 Added an INFRA ticket to do this here https://issues.apache.org/jira/browse/INFRA-16535
20. This was release as part of ORC 1.5.6.