Item 132
git_comments:

git_commits:

1. **summary:** [SPARK-11724][SQL] Change casting between int and timestamp to consistently treat int in seconds.
   message: [SPARK-11724][SQL] Change casting between int and timestamp to consistently treat int in seconds. Hive has since changed this behavior as well. https://issues.apache.org/jira/browse/HIVE-3454 Author: Nong Li <nong@databricks.com> Author: Nong Li <nongli@gmail.com> Author: Yin Huai <yhuai@databricks.com> Closes #9685 from nongli/spark-11724.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-11724] [SQL] Change casting between int and timestamp to consistently treat int in seconds.
   body: Hive has since changed this behavior as well. https://issues.apache.org/jira/browse/HIVE-3454

github_pulls_comments:

1. Good catch! The currently behaviour looks wierd to me, how does hive handle this? cc @yhuai
2. **[Test build #45842 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/45842/consoleFull)** for PR 9685 at commit [`9aa668c`]
   (https://github.com/apache/spark/commit/9aa668c4ec0b71a1e5177fc353fa21f40c0ee113). - This patch **fails Spark unit tests**. - This patch merges cleanly. - This patch adds the following public classes _(experimental)_:\n * `case class StddevSamp(child: Expression,`\n * `case class StddevPop(`\n * `class TypedColumn[-T, U](`\n
3. > Good catch! The currently behaviour looks wierd to me, how does hive handle this? cc @yhuai It looks like Hive fixed this in 1.2. https://issues.apache.org/jira/browse/HIVE-3454
4. Seems we need to update those golden files. I believe we can set `HIVE_HOME` to hive 1.2.1's dir and delete those golden files. Then, when we run those tests again, we will first let Hive generate results.
5. Seems Hive is still wrong when the data type is int. See ``` hive> SELECT CAST(CAST(-1200 AS TIMESTAMP) AS INT); OK -2 Time taken: 0.047 seconds, Fetched: 1 row(s) hive> SELECT CAST(CAST(-1200.0 AS TIMESTAMP) AS INT); OK -1200 Time taken: 0.046 seconds, Fetched: 1 row(s) ``` Let's delete those golden files and create our own test (using `test(...) {...}` instead of using `createQueryTest(..., ...)`).
6. **[Test build #46364 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/46364/consoleFull)** for PR 9685 at commit [`c4f2da1`]
   (https://github.com/apache/spark/commit/c4f2da13f0ebc39b5275f2a34bff0a5f95e66f6f). - This patch **fails from timeout after a configured wait of `250m`**. - This patch merges cleanly. - This patch adds no public classes.
7. test this please
8. **[Test build #46391 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/46391/consoleFull)** for PR 9685 at commit [`c4f2da1`]
   (https://github.com/apache/spark/commit/c4f2da13f0ebc39b5275f2a34bff0a5f95e66f6f). - This patch passes all tests. - This patch merges cleanly. - This patch adds no public classes.
9. **[Test build #46435 has finished]
   (https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/46435/consoleFull)** for PR 9685 at commit [`347de98`]
   (https://github.com/apache/spark/commit/347de98b9c91bda523f7dc04d4a272a2ed51842b). - This patch passes all tests. - This patch merges cleanly. - This patch adds no public classes.
10. LGTM. Thank you for the fix! Merging it to master and branch 1.6.

github_pulls_reviews:

1. Seems we do not include the new golden file? I can generate it tomorrow.

jira_issues:

1. **summary:** Problem with CAST(BIGINT as TIMESTAMP)
   description: Ran into an issue while working with timestamp conversion. CAST(unix_timestamp() as TIMESTAMP) should create a timestamp for the current time from the BIGINT returned by unix_timestamp() Instead, however, a 1970-01-16 timestamp is returned.

jira_issues_comments:

1. Additional information/workaround: This looks like a nanosecond bug similar to HIVE-3090, so I tried this: cast((unix_timestamp() * 1.0) as TIMESTAMP) and got this: 2012-09-17 14:49:39.0 Which is what it should be, but I would have expected this to be handled without the "conversion". cast(round(unix_timestamp(),3) as TIMESTAMP) Will also work.
2. unix_timestamp() returns seconds but Timestamp expects millisecond value if input type is integer or long. I think you should multiply the value with 1000 for proper converting.
3. These are both valid options to work around the bug. However from the Hive wiki: https://cwiki.apache.org/Hive/languagemanual-types.html#LanguageManualTypes-Timestamps "Supports traditional UNIX timestamp with optional nanosecond precision. Supported conversions: Integer numeric types: Interpreted as UNIX timestamp in seconds" The documentation is stating that TIMESTAMP is *supposed* to accept integer numeric types. Nanosecond is optional. I assume that the documentation in the wiki is the intended behavior. It isn't working as described in the wiki, that is why I have marked this as a bug.
4. You are right. It's not consistent with description in wiki. This should be fixed ASAP but backward compatibility also be considered. I don't know how to resolve.
5. After commenting on HIVE-3822 I've realized that even with this quasi-workaround, the cast(int/float/double as timestamp) is fundamentally broken. When an epoch (assumed GMT) timestamp is passed through the cast() function it is converted to a different time based on the local timezone...this doesn't happen if the timestamp is cast from a formatted date string....this behavior is inconsistent. Furthermore, when attempting to use the to_utc_timestamp() function with a epoch date value the implicit cast() poisons the result of the timestamp that is stored by to_utc_timestamp() Something is definitely wrong.
6. Using java.sql.Date instead of java.util.Date because java.sql.Date implements a timezone agnostic method of representing timestamps (always UTC, which is what a unix epoch timestamp is defined as)
7. I think a good approach would be to fix it in a way that it can work with both double and long values. This is what I did and works for me. Let me know what you think. In org.apache.hadoop.hive.serde2.objectinspector.primitive PrimitiveObjectInspectorUtils.java edit getTimestamp method and convert the long object to double. getTimestamp(Object o, PrimitiveObjectInspector oi) case LONG: //Timestamp conversion from LONG is messy. Converting to double. long tsLongPrimitive=((LongObjectInspector) oi).get(o); result =TimestampWritable.doubleToTimestamp(tsLongPrimitive*1.0); break;
8. Adding a patch for fixing the issue. Apply to trunk's serde/src/java/org/apache/hadoop/hive/serde2/objectinspector/primitive/PrimitiveObjectInspectorUtils.java.
9. Query : select cast(unix_timestamp()*1.0 as timestamp) , cast(unix_timestamp() as timestamp) from mycrime limit 1; Output(After patching with HIVE-3454.patch) - both returns same and correct timestamps. 2013-06-18 11:45:04 2013-06-18 11:45:04
10. Hi there, when is this patch going to be applied to trunk? it seems it's enough important issue to be included
11. {color:green}Overall{color}: +1 all checks pass Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12588389/HIVE-3454.patch {color:green}SUCCESS: {color} +1 4789 tests passed Test results: http://bigtop01.cloudera.org:8080/job/PreCommit-HIVE-Build/671/testReport Console output: http://bigtop01.cloudera.org:8080/job/PreCommit-HIVE-Build/671/console Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase {noformat} This message is automatically generated. ATTACHMENT ID: 12588389

12. Could this patch be checked in?
13. I see that this is still an issue. I found this today when trying to convert from integer to timestamp. In order to get this to work I had to add milliseconds (3 digits after the unix epoch) and also use a bigint intstead of int. I am using version 0.12. Any idea when this will be put in the main trunk?
14. I am using 0.12 release still this problems exists. Inconsistancy in casting. select cast(cast(cast(1000 as INT) as TIMESTAMP) as INT) from src limit 1; result: 1 select cast(cast(cast(1000 as BIGINT) as TIMESTAMP) as BIGINT) from src limit 1; result: 1 select cast(cast(cast(1000.0 as DOUBLE) as TIMESTAMP) as DOUBLE) from src limit 1; result: 1000.0 converting int/bigint/tinyint to timestamp and back to int/bigint/tinyint changes the value. int/bigint/tinyint/boolean conversion to timestamp, should take input in seconds, just like double. Issue raised in sep/2012, still problem persists. This requires a fix.
15. Update the primitive types including boolean, byte, short ,int, long to be consistently represent the time in seconds when we convert them into timestamp.
16. {color:red}Overall{color}: -1 at least one tests failed Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12697289/HIVE-3454.2.patch {color:red}ERROR: {color} -1 due to 6 failed/errored test(s), 7526 tests executed *Failed tests:* {noformat} TestCustomAuthentication - did not produce a TEST-*.xml file org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver_vectorization_decimal_date org.apache.hadoop.hive.cli.TestHBaseCliDriver.testCliDriver_hbase_timestamp org.apache.hadoop.hive.cli.TestMiniTezCliDriver.testCliDriver_vectorization_decimal_date org.apache.hadoop.hive.cli.TestMiniTezCliDriver.testCliDriver_vectorized_casts org.apache.hadoop.hive.cli.TestSparkCliDriver.testCliDriver_vectorization_decimal_date {noformat} Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2705/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2705/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-2705/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests exited with: TestsFailedException: 6 tests failed {noformat} This message is automatically generated. ATTACHMENT ID: 12697289 - PreCommit-HIVE-TRUNK-Build
17. Unit tests baseline updated.
18. {color:green}Overall{color}: +1 all checks pass Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12697536/HIVE-3454.3.patch {color:green}SUCCESS: {color} +1 7541 tests passed Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2732/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2732/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-2732/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase {noformat} This message is automatically generated. ATTACHMENT ID: 12697536 - PreCommit-HIVE-TRUNK-Build
19. +[~brocknoland] Can you take a look at the code?
20. Seems reasonable to me! [~spena] could you look as you have more experience here than I do.
21. We may need to mark this as an incompatible change - folks have been using this behavior (inconsistent as it is) for a while now. Also, HIVE-9298 added TimestampParser capable of interpreting numeric text input as milliseconds since Unix epoch. Should we change this to seconds to make it consistent with the changes done here?
22. [~jdere] Thanks for pointing that out. I marked it as incompatible change (of course technically it's not incompatible). Regarding the TimestampParser, since MillisDateFormatParser is just one of the parser to support, I think it should be good without any change. Probably we can support SecondsDateFormatParser in the future if it's necessary.
23. I hate new configuration...but if people are depending on this should we make the behavior change configurable?
24. Me either and I feel a new configuration may make things more complicated. From the comments above, actually the users are more using workarounds (converting int/bigint to float/double first before interpreting it as timestamp). My opinion is not to add a new configuration and keep things simple.
25. Me either and I feel a new configuration may make things more complicated. From the comments above, actually the users are more using workarounds (converting int/bigint to float/double first before interpreting it as timestamp). My opinion is not to add a new configuration and keep things simple.

26. It looks very good. I agree with [~aihuaxu], if users were using workarounds, then they shouldn't have any problem with this change, as float/double would work fine.The only thing is what if users are using integer values to get the timestamp? If they know that they need to pass milliseconds, they they could have these values in their queries; and now this change would give them unexpected values. However, the wiki states that seconds is the allowed number, so these users are using it incorrectly. I agree with this change so that it fixes the bug, and complies with the wiki page. +1
27. Discussed with Brock offline, seems more reasonable to make it configurable so that we won't break existing customers. Set "int.timestamp.conversion.in.seconds" to true in hive-site.xml to make the behavior consistent. We will gradually switch to the seconds interpretation.
28. Can you make the naming of the HiveConf parameter more consistent with the rest of the HiveConf variable names (starts with "hive.").
29. {color:green}Overall{color}: +1 all checks pass Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12699343/HIVE-3454.3.patch {color:green}SUCCESS: {color} +1 7553 tests passed Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2817/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2817/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-2817/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase {noformat} This message is automatically generated. ATTACHMENT ID: 12699343 - PreCommit-HIVE-TRUNK-Build
30. Thanks [~jdere] for reviewing. Just updated the parameter name.
31. Thanks [~jdere] for reviewing. Just updated the parameter name.
32. {color:red}Overall{color}: -1 at least one tests failed Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12699474/HIVE-3454.3.patch {color:red}ERROR: {color} -1 due to 1 failed/errored test(s), 7557 tests executed *Failed tests:* {noformat} org.apache.hadoop.hive.cli.TestMinimrCliDriver.testCliDriver_schemeAuthority {noformat} Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2820/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2820/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-2820/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests exited with: TestsFailedException: 1 tests failed {noformat} This message is automatically generated. ATTACHMENT ID: 12699474 - PreCommit-HIVE-TRUNK-Build
33. The test failure is unrelated to the change.
34. Have we tested this as part of an MR job? I don't think that the hive-site.xml is shipped as part of MR jobs. If that is true, how about we do as follows: 1) Add method {{public static void initialize(Configuration)}} to {{TimestampWritable}} 2) Call this method from {{AbstractSerDe.initialize}} which should be called, with configuration, in all the right places. 3) In {{TimestampWritable.initialize}} you can use the static {{HiveConf.getBoolVar}} a bit kludgy but it should work. This all assuming the current impl doesn't work. bq. "timestamp conversion." I think we need a space after this.
35. Yeah. I have tested with an MR job and it picks up the hive-site.xml without the problem with hiveserver2 or CLI.
36. If this config setting is initialized once in a static block, then for hiveserver2 all subsequent sessions would be stuck with the initial setting regardless of the config settings of the session, right? During the MR jobs, would we then see the sec/msec behavior flip to use the session's config settings since the static variable is being initialized for the first time in MR task?
37. Yeah. Thanks for pointing that out. I also notice that problem. We won't be able to override during the session with that approach and also we could have problem if we run in the real cluster as Brock pointed out. I'm trying a different approach.
38. Read the configuration from passed in Configuration object from the initialize() in AbstractSerDe class. Since AvroSerDe overrides the function and OrcSerDe doesn't inherit from AbstractSerDe right now, added in these two classes as well. It will support the session override.
39. {color:red}Overall{color}: -1 at least one tests failed Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12699952/HIVE-3454.4.patch {color:red}ERROR: {color} -1 due to 1 failed/errored test(s), 7568 tests executed *Failed tests:* {noformat}

org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver_udaf_percentile_approx_23 {noformat} Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2836/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2836/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-2836/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests exited with: TestsFailedException: 1 tests failed {noformat} This message is automatically generated. ATTACHMENT ID: 12699952 - PreCommit-HIVE-TRUNK-Build
40. +1 LGTM best we can do in this situation.
41. TimestampWritable.intToTimeStampInSeconds is still static, which means we could run into issues in HiveServer2 with concurrent queries. Maybe this should be thread-local. Hmm, yeah I'm not sure where the best place is to call TimestampWritable.initialize() .. I was going to recommend Driver.compile(), though we would also have to add it to MR task initialization somewhere if there is such a place.
42. [~jdere] I spent some time looking into the parallel in Hive. Hive can handle queries in parallel or a complex query breaking into subtasks executed in parallel. In both cases, Hive hands over the multiple jobs to hadoop and those jobs won't affect each other. I have simulated the cases by debugging them from different sessions and it works as expected.
43. The unit test failure is unrelated to the change. [~jdere] More comments on my input above?
44. I guess the only possible issue here might be for expressions that get evaluated during compilation as opposed to during the tasks, something like cast(0 as timestamp) which would get evaluated by constant folding. Not sure if there is any other possible work/tasks that happen locally in-process.
45. If the job runs locally in-process, like cast(0 as timestamp), SerDe is still used for intermediate output and thus the property still gets set and works fine.
46. If concurrent compilation is possible, then there is still the possibility of multiple threads setting/accessing the same static variable and messing up the config setting for the other threads. You're adding this to TimestampWritable - there is even a member in that class that was changed to thread-local as a result of HIVE-4516.
47. I see your points that multiple threads run in single mapper or reducer access that class and that may cause concurrency issue. I will try to call the initialize() from Mapper and Reducer's initialization function to make sure it initializes once. And also Driver.compile() (I will check what would be the good place for the local mode).
48. Updated to initialize for local mode in Driver.runInternal() and configure() of ExecMapper and ExecReducer.
49. {color:green}Overall{color}: +1 all checks pass Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12701998/HIVE-3454.4.patch {color:green}SUCCESS: {color} +1 7580 tests passed Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2921/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/2921/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-2921/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase {noformat} This message is automatically generated. ATTACHMENT ID: 12701998 - PreCommit-HIVE-TRUNK-Build
50. [~jdere] Can you review the latest patch? Right now it should be set once for each Mapper and Reducer.
51. - Can you make intToTimeStampInSeconds thread-local? - According to [~vikram.dixit], Tez uses separate classes, the equivalent to ExecMapper/ExecReducer are MapRecordProcessor/ReduceRecordProcessor. It looks like Spark also has its own equivalents (SparkMapRecordHandler/SparkMapReduceHandler). If you want we can defer those changes and open a follow-up Jira to do that work.
52. #Actually we don't want it to be thread-local since we want all threads, e.g, within Mapper to have the same settings. If we set to be thread-local, the threads within the same mapper which don't set intToTimeStampInSeconds will have incorrect behaviors. Of course, the original problem was that we can't access hive properties from TimestampWritable class and we have to set from outside. #I will create a follow-up jira to do the rest (Tez and Spark).
53. Please don't use the static intToTimeStampInSeconds, it is not thread-safe
54. {quote} Actually we don't want it to be thread-local since we want all threads, e.g, within Mapper to have the same settings. If we set to be thread-local, the threads within the same mapper which don't set

intToTimeStampInSeconds will have incorrect behaviors. Of course, the original problem was that we can't access hive properties from TimestampWritable class and we have to set from outside. {quote} Asking around a bit about HIVE-7926, it sounds like LLAP will potentially have multiple queries running at the same time in different threads, which would make a single static variable problematic. It also sounds like each thread should in theory have the equivalent of something like ExecMapper.configure() where thread-local context is initialized, so I think that any threads running map/reduce work should have an opportunity to set intToTimeStampInSeconds. I would think that things would be similar for Spark as well, though someone with more familiarity with Hive-Spark execution would be better suited to answer that question.

55. HOS won't have this issue as the executors are tied to a user session. I think we should commit the patch as is and create a follow-up to address the issue with LLAP.
56. I'm willing to defer that work to a different Jira. +1 But I'm surprised this would not be an issue with HOS as well .. can HOS have multiple threads doing work for more than one query at the same time? If so then HOS would get hit by this too since this setting is relying on a static variable.
57. bq. an HOS have multiple threads doing work for more than one query at the same time? no it cannot. HOS does do concurrent tasks but not across queries.
58. Was not aware of LLAP. Seems reasonable to defer the issue since I guess LLAP needs to handle many other similar thread-safe problems like this. Correct me if I'm wrong with the hadoop execution: currently (without LLAP) each ExecMapper and ExecReducer runs in its own JVM and it could spin off new threads. Each ExecMapper and ExecReducer handles one job from a query and we want those threads to share that same static variable value. With LLAP, if ExecMapper runs in a thread and doesn't spin off new threads, then thread-local would work, but we still need to handle current existing scenario. If new threads are spun off, thread-local actually won't work.
59. It just seems like a bad design pattern irrespective of LLAP. [~hagleitn] did some work in operator pipeline to clean up statics and such, he can comment. What is the real scope of this variable? It should be scoped accordingly. Even a singleton map would be better...
60. I agree with [~sershe]. These statics aren't just a pain w/ multi threading, they also hurt if you just re-use the same jvm for multiple things. This can happen in tez, spark and if we decide to run stuff on the client/in hs2 afaik. More specifically though: This config should take effect at compile time. There is no reason to evaluate the condition for each value in each row at run time. We should be able to just install the appropriate udf when we compile the query, no?
61. I agree that for non thread-safe statics, it will cause problems for multi-threading, while for constants or immutable objects, I guess you would share across threads rather than creating each copy for each thread. This is the case of a java primitive that we initialize once and use/read later by threads. Regarding UDF approach, I looked into that and it seems promising, while the current approach actually only evaluates the value when the type is timestamp. I guess it would be the same as UDF approach.
62. Of course, since we need to support LLAP, UDF approach seems to be the right approach.
63. I'm looking into the solution [~hagleitn] suggested. I will break the tasks into separate tasks, so that this jira will focus on the right/simple behavior and a followup to focus on making it configurable so that we can easily revert in the future and also focus on that only.
64. [~brocknoland], [~jdere] and [~hagleitn] I created HIVE-9917 for the followup change. I will mark the current change as incompatible. How does that sound?
65. +1. Breaking into separate tasks makes sense to me, but it's better to have others' inputs (I'm no expert on this matter), before we commit this.
66. That's fine to break into separate tasks, we'll just need to make sure the followup config task gets in before the next release.
67. This has been marked patch available - which one should we be looking at - HIVE-3454.3.patch?
68. Yes. That's right. That patch is to change in the correct way to interpreting all the datatypes as seconds. Thanks for looking.
69. I've just started looking at vectorized code .. looks like the corresponding change to make for the vectorized path will be in MathExpr.doubleToTimestamp(). CC'ing [~mmccline] in case there is any more details to add here.
70. Thanks [~jdere]. We need to correct MathErpr.longToTimestamp() function since we have inconsistency with the int/long type, Correct?
71. Whoops, yes it is longToTimestamp() that should be fixed, since the point of this is to correct the int/long to timestamp behavior.
72. Made additional changes to vectorized functions and unit tests.
73. {color:red}Overall{color}: -1 at least one tests failed Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12705949/HIVE-3454.3.patch {color:red}ERROR:

{color} -1 due to 1 failed/errored test(s), 7820 tests executed *Failed tests:* {noformat} org.apache.hive.jdbc.TestMultiSessionsHS2WithLocalClusterSpark.testSparkQuery {noformat} Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/3091/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/3091/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-3091/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests exited with: TestsFailedException: 1 tests failed {noformat} This message is automatically generated. ATTACHMENT ID: 12705949 - PreCommit-HIVE-TRUNK-Build

74. So this patch did not require any changes to MathExpr? How does the behavior of the vectorized cast change?
75. Attached should be the right patch now. Maybe I uploaded the right patch first but I worried test run was not started (but seems the test was run against the right patch), so I uploaded again but the wrong one. Thanks for catching it.
76. {color:red}Overall{color}: -1 at least one tests failed Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12706162/HIVE-3454.3.patch {color:red}ERROR: {color} -1 due to 3 failed/errored test(s), 7819 tests executed *Failed tests:* {noformat} org.apache.hadoop.hive.cli.TestMiniTezCliDriver.testCliDriver_vector_between_in org.apache.hadoop.hive.ql.exec.vector.expressions.TestVectorTypeCasts.testCastLongToTimestamp org.apache.hive.jdbc.TestMultiSessionsHS2WithLocalClusterSpark.testSparkQuery {noformat} Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/3105/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/3105/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-3105/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests exited with: TestsFailedException: 3 tests failed {noformat} This message is automatically generated. ATTACHMENT ID: 12706162 - PreCommit-HIVE-TRUNK-Build
77. {color:red}Overall{color}: -1 at least one tests failed Here are the results of testing the latest attachment: https://issues.apache.org/jira/secure/attachment/12706362/HIVE-3454.3.patch {color:red}ERROR: {color} -1 due to 1 failed/errored test(s), 7819 tests executed *Failed tests:* {noformat} org.apache.hadoop.hive.cli.TestCliDriver.testCliDriver_udaf_percentile_approx_23 {noformat} Test results: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/3107/testReport Console output: http://ec2-174-129-184-35.compute-1.amazonaws.com/jenkins/job/PreCommit-HIVE-TRUNK-Build/3107/console Test logs: http://ec2-174-129-184-35.compute-1.amazonaws.com/logs/PreCommit-HIVE-TRUNK-Build-3107/ Messages: {noformat} Executing org.apache.hive.ptest.execution.PrepPhase Executing org.apache.hive.ptest.execution.ExecutionPhase Executing org.apache.hive.ptest.execution.ReportingPhase Tests exited with: TestsFailedException: 1 tests failed {noformat} This message is automatically generated. ATTACHMENT ID: 12706362 - PreCommit-HIVE-TRUNK-Build
78. This error is unrelated to the change and seems random.
79. +1
80. Committed to trunk. Thanks Aihua!
81. Thanks [~csun] and [~jdere].
82. This issue has been fixed and released as part of the 1.2.0 release. If you find an issue which seems to be related to this one, please create a new jira and link this one with new jira.
83. Seems this issue has not been fixe completely. I still see {code} hive> SELECT CAST(CAST(-1200.0 AS TIMESTAMP) AS DOUBLE); OK -1200.0 Time taken: 0.047 seconds, Fetched: 1 row(s) hive> SELECT CAST(CAST(-1200 AS TIMESTAMP) AS INT); OK -2 Time taken: 0.044 seconds, Fetched: 1 row(s) {code}
84. [~yhuai] Which version are you running on? I tried on the latest and it seems to be fine.
85. I was using 1.2.1.
86. [~yhuai] In 1.2.1, you need to {{set hive.int.timestamp.conversion.in.seconds=true;}} to get the correct behavior, see HIVE-9917. We kept the existing behavior for backward compatibility. In 2.0.0, we default hive.int.timestamp.conversion.in.seconds to true.

87. OK. Thanks.