

**git\_comments:**

1. it's possible a partition exists in metastore but the data has been removed
2. remove one partition

**git\_commits:**

1. **summary:** [FLINK-16197][hive] Failed to query partitioned table when partition folder is removed  
**message:** [FLINK-16197][hive] Failed to query partitioned table when partition folder is removed This closes #11175

**github\_issues:****github\_issues\_comments:****github\_pulls:**

1. **title:** [FLINK-16197][hive] Failed to query partitioned table when partition ...  
**body:** ...folder is removed <!-- \*Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.\* \*Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.\* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where \*FLINK-XXXX\* should be replaced by the actual issue number. Skip \*component\* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide](<https://flink.apache.org/contributing/contribute-code.html#open-a-pull-request>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. \*\* (The sections below can be removed for hotfixes of typos)\*\* --> ## What is the purpose of the change If a partition exists in HMS but not in HDFS, the query fails with `` Caused by: org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: hdfs://..... at org.apache.hadoop.mapred.LocatedFileStatusFetcher.getFileStatuses(LocatedFileStatusFetcher.java:155) at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:237) at org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:315) at org.apache.flink.connectors.hive.read.HiveTableInputFormat.createInputSplits(HiveTableInputFormat.java:219) at org.apache.flink.connectors.hive.HiveTableSource.getDataStream(HiveTableSource.java:152) `` Expected behavior is to run the query ignoring this partition. ## Brief change log - When generating input splits, skip a partition if the data folder doesn't exist. - Add test ## Verifying this change Added test case ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? NA

**github\_pulls\_comments:**

1. Thanks a lot for your contribution to the Apache Flink project. I'm the @flinkbot. I help the community to review your pull request. We will use this comment to track the progress of the review. ## Automated Checks Last check on commit 53fda3bf9cc936fee8c1925c76730df8d9ee6381 (Fri Feb 28 21:48:26 UTC 2020)

- \*\*Warnings:\*\*** \* No documentation files were touched! Remember to keep the Flink docs up to date! \* \*\*This pull request references an unassigned [Jira ticket](https://issues.apache.org/jira/browse/FLINK-16197). \*\*
- According to the [code contribution guide](https://flink.apache.org/contributing/contribute-code.html), tickets need to be assigned before starting with the implementation work. <sub>Mention the bot in a comment to re-run the automated checks.</sub> ## Review Progress \* ? 1. The [description] looks good. \* ? 2. There is [consensus] that the contribution should go into to Flink. \* ? 3. Needs [attention] from. \* ? 4. The change fits into the overall [architecture]. \* ? 5. Overall code [quality] is good. Please see the [Pull Request Review Guide](https://flink.apache.org/contributing/reviewing-prs.html) for a full explanation of the review process. <details> The Bot is tracking the review progress through labels. Labels are applied according to the order of the review items. For consensus, approval by a Flink committer or PMC member is required <summary>Bot commands</summary> The @flinkbot bot supports the following commands: - `@flinkbot approve description` to approve one or more aspects (aspects: `description`, `consensus`, `architecture` and `quality`) - `@flinkbot approve all` to approve all aspects - `@flinkbot approve-until architecture` to approve everything until `architecture` - `@flinkbot attention @username1 [@username2 ..]` to require somebody's attention - `@flinkbot disapprove architecture` to remove an approval you gave earlier </details>
2. @JingsongLi @bowenli86 Let me know if you think the change makes sense. Thanks,
  3. Hi @lirui-apache , can you describe the issue details in pull request too?
  4. Is this a reasonable issue?
  5. <!-- Meta data { "version" : 1, "metaDataEntries" : [ { "hash" : "53fda3bf9cc936fee8c1925c76730df8d9ee6381", "status" : "DELETED", "url" : "https://travis-ci.com/flink-ci/flink/builds/150017255", "triggerID" : "53fda3bf9cc936fee8c1925c76730df8d9ee6381", "triggerType" : "PUSH" }, { "hash" : "8aedd02e74d1a07be08e367b693ad82d69bfde3b", "status" : "DELETED", "url" : "https://travis-ci.com/github/flink-ci/flink/builds/154595008", "triggerID" : "8aedd02e74d1a07be08e367b693ad82d69bfde3b", "triggerType" : "PUSH" }, { "hash" : "8aedd02e74d1a07be08e367b693ad82d69bfde3b", "status" : "DELETED", "url" : "https://dev.azure.com/rmetzger/5bd3ef0a-4359-41af-abca-811b04098d2e/\_build/results?buildId=6536", "triggerID" : "8aedd02e74d1a07be08e367b693ad82d69bfde3b", "triggerType" : "PUSH" }, { "hash" : "f41f4359a68f8c9b85a33d3414bf346e02c17d6a", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=1842", "triggerID" : "f41f4359a68f8c9b85a33d3414bf346e02c17d6a", "triggerType" : "PUSH" }, { "hash" : "7cf8bc2371f60ce02daec08bda96b30e8ab94a32", "status" : "SUCCESS", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=1900", "triggerID" : "7cf8bc2371f60ce02daec08bda96b30e8ab94a32", "triggerType" : "PUSH" } ] --> ## CI report: \* 7cf8bc2371f60ce02daec08bda96b30e8ab94a32 Azure: [SUCCESS](https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=1900) <details> <summary>Bot commands</summary> The @flinkbot bot supports the following commands: - `@flinkbot run travis` re-run the last Travis build - `@flinkbot run azure` re-run the last Azure build </details>
  6. > Is this a reasonable issue? The issue was reported by a user trying to access Hive tables from Flink. Since Hive can handle this situation, I think a migrating user might expect Flink to do the same. BTW, Hive handles the issue a little differently. Instead of ignoring the missing folder, it creates empty files/folders for such partitions, under a scratch dir in HDFS. Since we don't have such a scratch dir on Flink side (therefore unable to do the clean-up), I think we can't follow Hive's solution. Also note that this is not trying to deal with concurrency issues. Concurrently accessing the table (e.g. delete the files while another job reads it) can still fail.
  7. Also cc @zjffdu
  8. @JingsongLi This is a real use case that one user hit when using Flink on Zeppelin. This happens when the data on hdfs is inconsistent with hive metadata.
  9. feels like Hive's approach is more graceful, though not perfect, as it still respect HMS as the source of truth. This fix would make Flink disrespect any discrepancy between metadata in HMS and actual storage. That seems to minimize migration cost, but I'm not sure that's a good longterm strategy
  10. Hi @bowenli86 , could you elaborate why it's not a good longterm strategy? IMHO, it should be up to each connector to decide how to handle discrepancy between metadata and storage. Because such discrepancies might be treated differently in the external systems. One system may consider the discrepancy as a fatal error, and another system may expect the discrepancy to happen from time to time and choose to tolerate it. Therefore I think each connector should follow the behavior of the external system it connects to.
  11. there shouldn't be discrepancy in HMS and hdfs in the first place, users need to figure out what led to such broken state and fix it. In this case, shall the right approach be that user just drop the partition whose dir has been removed already?
  12. @bowenli86 I agree we should educate users to avoid the discrepancy in the first place. Hive provides the `MSCK` tool so that users can sync partition info between metadata and storage by running this command. But

- I think it's one thing to tell users the best practice, and another to be robust when something is off. Users migrating to Flink would find it more friendly if they can have the same level of robustness.
13. I have a slightly different opinion on this. Though it might mitigate the problem for users, we are indeed trying to hack the way around invalid inputs passed to our APIs. E.g. flink's file source will fail if the target is missing, rather than mitigate for users. Maybe @KurtYoung or @JingsongLi can give some opinions and help to merge if they think this is a proper fix?
  14. I understand Flink may handle such issues differently for its own file sources. But for Hive connector, metadata is retrieved from (and managed by) an external HMS. If Hive, the owner of the metadata, is aware of such discrepancy and decides it should be tolerated, I don't see a good reason why we should be more strict on that matter.
  15. If i understand this correctly, ignore the missing partition doesn't indicate flink disrespect HMS as source of truth. I think it's important to figure out what does it mean for a partition without any directory or files. It looks to me hive treat it as an empty partition, because it not only ignore the error but also helped to fix this partition by creating an empty directory (correct me if i'm wrong). If this is true, then just ignoring it in Flink sounds reasonable to me. Ignoring this inconsistent means flink also treat it as an empty partition, but we don't have the ability to fix it.
  16. @KurtYoung You're right that Hive treats it as an empty partition. But Hive doesn't fix it. The empty folder/file is created in some scratch dir, not under the table location. I guess it's just meant to give the input format something to work on.
  17. @lirui-apache Then this approach sounds good to me. @bowenli86 do you have any further concerns?
  18. Any progress? @lirui-apache Can you update the PR? If there is no objection, I think we can merge this.

#### github\_pulls\_reviews:

#### jira\_issues:

1. **summary:** Failed to query partitioned table when partition folder is removed  
**description:** If a partition exists in HMS but not in HDFS, the query fails with {noformat} Caused by:  
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: hdfs://..... at  
org.apache.hadoop.mapred.LocatedFileStatusFetcher.getFileStatuses(LocatedFileStatusFetcher.java:155) at  
org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:237) at  
org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:315) at  
org.apache.flink.connectors.hive.read.HiveTableInputFormat.createInputSplits(HiveTableInputFormat.java:219)  
at org.apache.flink.connectors.hive.HiveTableSource.getDataStream(HiveTableSource.java:152) {noformat}  
Expected behavior is to run the query ignoring this partition.

#### jira\_issues\_comments:

1. master: e8767af47f1d85318ee4853f678df24075e2e794 release-1.11: bdffe91c6ae81a2cdb6033d241946c8d09b96c7c
2. [~lirui] If you think it is worth to finish in 1.10 too, please re-open this and submit a PR.