Item 256

git_comments:

git_commits:

1. **summary:** HIVE-1001 CombinedHiveInputFormat should parse the inputpath correctly
   message: HIVE-1001 CombinedHiveInputFormat should parse the inputpath correctly git-svn-id: https://svn.apache.org/repos/asf/hadoop/hive/trunk@893029 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** CombinedHiveInputFormat should parse the inputpath correctly
   description: From David Lerman: " I'm running into errors where CombinedHiveInputFormat is combining data from two different tables which is causing problems because the tables have different input formats. It looks like the problem is in org.apache.hadoop.hive.shims.Hadoop20Shims.getInputPathsShim. It calls CombineFileInputFormat.getInputPaths which returns the list of input paths and then chops off the first 5 characters to remove file: from the beginning, but the return value I'm getting from getInputPaths is actually hdfs://domain/path. So then when it creates the pools using these paths, none of the input paths match the pools (since they're just the file path which protocol or domain). " We should use Path.getPath() to get the path part of an URI instead of just chopping off 5 chars.

jira_issues_comments:

1. This is a blocker for 0.5
2. +1. Looks good. Will commit if tests pass.
3. I just committed. Thanks Namit!
4. **body:** This patch appears to affect other functionality. Without the patch, "insert overwrite directory 'out' select ... from ..." yields 1 MR job, but with the patch, it yields 2 MR jobs. At first glance, it looks like the second job in the insert overwrite plan is a conditional operation where it either does an HDFS move to move the data from the temp output directory to its final destination, or runs a MR job to copy the data (maybe if it needs to copy across clusters?). Before the patch, it just does the HDFS copy, but after it, it does the MR copy. Maybe the paths its comparing to determine which task to run are getting screwed up by the patch? Steps to reproduce: Using Cloudera's Hadoop 0.20.1+152 (since the CombineFile functionality doesn't work in 0.20.1 without the extra patches): * Create a data file twolines.dat containing: key1^Avalue1 key2^Avalue2 * Create a table with two partitions each containing that data: CREATE TABLE fourlinestest(KEY STRING, VALUE STRING) PARTITIONED BY (part int) STORED AS TEXTFILE; load data local inpath 'twolines.dat' into table fourlinestest partition (part=1); load data local inpath 'twolines.dat' into table fourlinestest partition (part=2); * Using Hive r888452 (before this patch was applied): set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat; insert overwrite directory 'out' select key from fourlinestest; --> The log shows that that the pools are getting created with corrupt paths (which was the bug that was fixed), but the job runs successfully with 1 MR job: Total MapReduce jobs = 2 Launching Job 1 out of 2 Number of reduce tasks is set to 0 since there's no reduce operator Starting Job = job_200912240057_7597, Tracking URL = http://REMOVED:50030/jobdetails.jsp?jobid=job_200912240057_7597 Kill Command = REMOVED/bin/hadoop job -Dmapred.job.tracker=REMOVED:8021 -kill job_200912240057_7597 2009-12-28 14:47:02,350 Stage-1 map = 0%, reduce = 0% 2009-12-28 14:47:14,189 Stage-1 map = 100%, reduce = 0% 2009-12-28 14:47:18,404 Stage-1 map = 100%, reduce = 100% Ended Job = job_200912240057_7597 Launching Job

2 out of 2 Moving data to: hdfs://REMOVED/561636114/10000 Moving data to: out 4 Rows loaded to out OK * Apply the patch, rebuild, and rerun patch -p0 < hive.1001.1.patch ant package bin/hive set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat; insert overwrite directory 'out' select key from fourlinestest; --> This time the second job actually runs a MR job (which then fails because of HIVE-1006). Total MapReduce jobs = 2 Launching Job 1 out of 2 Number of reduce tasks is set to 0 since there's no reduce operator Starting Job = job_200912240057_7616, Tracking URL = http://REMOVED:50030/jobdetails.jsp?jobid=job_200912240057_7616 Kill Command = REMOVED/bin/hadoop job -Dmapred.job.tracker=REMOVED:8021 -kill job_200912240057_7616 2009-12-28 14:54:39,414 Stage-1 map = 0%, reduce = 0% 2009-12-28 14:54:51,224 Stage-1 map = 100%, reduce = 0% Ended Job = job_200912240057_7616 Launching Job 2 out of 2 Number of reduce tasks determined at compile time: 1 In order to change the average load for a reducer (in bytes): set hive.exec.reducers.bytes.per.reducer=<number> In order to limit the maximum number of reducers: set hive.exec.reducers.max=<number> In order to set a constant number of reducers: set mapred.reduce.tasks=<number> java.io.IOException: cannot find dir = hdfs://RMOVED/1656362434/10001/attempt_200912240057_7616_m_000000_0 in partToPartitionInfo!
label: code-design

5. Hi Dave, How many mappers in the map-reduce job before applying this patch? It is strange that the merge job was not running. The merge job is started according to average result file size produced by the first job, and the merge job always uses a map-reduce job. And also discussed with Namit, we may need to add another parameter "number of files" to determine whether to start the merge job or not (this will be a different issue).
6. Okay, that makes sense then. Without the patch, the two input pools have corrupt paths so the input files don't match either pool and get processed together in one pool of non-matching paths. This yields one split and one mapper, so the merge step doesn't run (since there's only one output file). With the patch, the pools get created correctly, so the two files are processed in separate pools, which yields two splits and two mappers, so the merge step runs. Thanks for the help.
7. Closing the jira as per the clarification