Item 213
git_comments:

git_commits:

1. **summary:** Merge pull request #1 from jalkjaer/cascading_sink
   message: Merge pull request #1 from jalkjaer/cascading_sink NULL tuples causes NPE when writing

github_issues:

github_issues_comments:

github_pulls:

1. **title:** Filtering records across multiple blocks
   body: Copied from: https://github.com/Parquet/parquet-mr/pull/413 However, as tomwhite mentioned, there might be a better way to do this. I had also written this: `current ++;` still doesn't seem correct even when `currentValue != null`. Imagine a block with 100 records, but only the record at position 50 matches our filter. In this case, the first time `nextKeyValue()` is called, it will call `recordReader.read()` which will successfully find the record at pos 50, but `current` will just be incremented to 1.

github_pulls_comments:

1. @onlynone, I agree with you, however I think the fix is still functionally correct. That's what I meant about ensuring `getProgress()` is correct - although since it is used to give a rough measure of MR progress this change doesn't break applications, it just underestimates progress. Having said that, here's another fix that correctly updates `current`: https://github.com/tomwhite/parquet-mr/compare/pr-413-change-filtering
2. I think that Tom's fix is correct and that's a reasonable work-around for right now. But I'd rather get rid of the recursive call because that will increase the stack for each filtered record. Here's a version that just loops until the internal reader starts returning non-null records again. It also checks to make sure the total isn't going past the currently loaded limit so that there aren't conditions where it would loop infinitely. ``` java try { checkRead(); currentValue = recordReader.read(); current ++; // only happens with FilteredRecordReader at end of block while (currentValue == null && current < total && current <= totalCountLoadedSoFar) { checkRead(); currentValue = recordReader.read(); current ++; } if (DEBUG) LOG.debug("read value: " + currentValue); } catch (RuntimeException e) { throw new ParquetDecodingException(format("Can not read value at %d in block %d in file %s", current, currentBlock, file), e); } ``` Like you said, a real fix needs to correctly keep track of the records that are filtered out. How about adding a count accessor to parquet.io.RecordReader? That would be a quick fix, but I'd rather see a better contract with the record reader that strictly defines behavior when it runs out of records and maybe keeps track internally. Iterator is good inspiration.
3. Thanks for the review @rdblue. I agree that the minimal fix is the way to go to get this fixed in the short term; for one thing changing (Filtered)RecordReader causes the semantic versioning plugin to complain. I've updated the minimal fix to avoid the recursive call as you suggested. See https://github.com/apache/incubator-parquet-mr/pull/9. It's slightly different to your code since we need to take account of the case where there are no further non-null records - i.e. the while loop needs to return false for that case. I've added a test for that case and also for the case where only the last block has a record that matches the filter.
4. LGTM. Could you Open a parquet JIRA and prefix the name of the PR with its ID as described in the following link ? https://github.com/apache/incubator-parquet-mr/pull/8/files?short_path=6a33714#diff-6a3371457528722a734f3c51d9238c13
5. Thanks for taking a look, Julien. I've opened [PARQUET-9](https://issues.apache.org/jira/browse/PARQUET-9) for this.
6. Was this included in https://github.com/apache/incubator-parquet-mr/pull/9 ?
7. @julienledem: yes. I think Tom had to create a new pull request because he couldn't push review changes to this one.
8. This is fixed: https://github.com/apache/incubator-parquet-mr/commit/2d8ebdbe00786823658bcdd2817e6b5afee15b25 @onlynone could you close this pull request?

9. Thanks Guys!
10. Thank you @onlynone !

github_pulls_reviews:

jira_issues:

jira_issues_comments: