Item 334
git_comments:

git_commits:

1. **summary:** [CARBONDATA-3026] clear expired property that may cause GC problem
   message: [CARBONDATA-3026] clear expired property that may cause GC problem During data loading, we will write some temp files (sort temp files and temp fact data files) in some locations. In currently implementation, we will add the locations to the CarbonProperties and associated it with a special key that refers to the data loading. After data loading, the temp locations are cleared, but the added property is still remain in the CarbonProperties and never to be cleared. This will cause the CarbonProperties object growing bigger and bigger and lead to OOM problems if the thrift-server is a long time running service. A local test shows that after adding different properties for 11 Billion times, the OOM happens. In this commit, I clear the property for the locations when we clear the locations. This closes #2833

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [CARBONDATA-3026] clear expired property that may cause GC problem
   body: During data loading, we will write some temp files (sort temp files and temp fact data files) in some locations. In currently implementation, we will add the locations to the CarbonProperties and associated it with a special key that refers to the data loading. After data loading, the temp locations are cleared, but the added property is still remain in the CarbonProperties and never to be cleared. This will cause the CarbonProperties object growing bigger and bigger and lead to OOM problems if the thrift-server is a long time running service. A local test shows that after adding different properties for 11 Million times, the OOM happens. In this commit, I clear the property for the locations when we clear the locations. Be sure to do all of the following checklist to help us incorporate your contribution quickly and easily: - [x] Any interfaces changed? `NO` - [x] Any backward compatibility impacted? `NO` - [x] Document update required? `NO` - [x] Testing done Please provide details on - Whether new unit test cases have been added or why no new tests are required? `NO` - How it is tested? Please attach test report. - Is it a performance related change? Please attach the performance test report. `NO` - Any additional information to help reviewers in testing this change. `NO` - [x] For large changes, please consider breaking it into sub-tasks under an umbrella JIRA. `NA`
   label: code-design

github_pulls_comments:

1. Build Success with Spark 2.1.0, Please check CI http://136.243.101.176:8080/job/ApacheCarbonPRBuilder2.1/848/
2. Build Failed with Spark 2.2.1, Please check CI http://95.216.28.178:8080/job/ApacheCarbonPRBuilder1/1045/
3. Build Failed with Spark 2.3.1, Please check CI http://136.243.101.176:8080/job/carbondataprbuilder2.3/9113/
4. Is the CI failure related to this PR?
5. yes. I add another commit to fix this error. ``` For a CarbonWriter, it will clean the temp store at closure. If we clean it for multiple times, error will occurs, because the corresponding property key has already been cleaned. In this commit, we remove the unnecessary clean outside the closure. ```
6. Build Success with Spark 2.1.0, Please check CI http://136.243.101.176:8080/job/ApacheCarbonPRBuilder2.1/852/
7. Build Failed with Spark 2.2.1, Please check CI http://95.216.28.178:8080/job/ApacheCarbonPRBuilder1/1049/
8. Build Success with Spark 2.3.1, Please check CI http://136.243.101.176:8080/job/carbondataprbuilder2.3/9117/
9. retest this please

10. Build Success with Spark 2.1.0, Please check CI
    http://136.243.101.176:8080/job/ApacheCarbonPRBuilder2.1/865/
11. Build Success with Spark 2.2.1, Please check CI
    http://95.216.28.178:8080/job/ApacheCarbonPRBuilder1/1062/
12. Build Success with Spark 2.3.1, Please check CI
    http://136.243.101.176:8080/job/carbondataprbuilder2.3/9130/
13. LGTM

github_pulls_reviews:

1. FYI: the clean operation will be done at the closure. there is no need to do this ahead

jira_issues:

1. **summary:** clear expired property that may cause GC problem
   description: During data loading, we will write some temp files (sort temp files and temp fact data files) in some locations. In currently implementation, we will add the locations to the CarbonProperties and associated it with a special key that refers to the data loading. After data loading, the temp locations are cleared, but the added property is still remain in the CarbonProperties and never to be cleared. This will cause the CarbonProperties object growing bigger and bigger and lead to OOM problems if the thrift-server is a long time running service. A local test shows that after adding different properties for 11 Billion times, the OOM happens.

jira_issues_comments: