Item 301
git_comments:

1. buffered reader
2. Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
3. not support
4. EOF
5. if requested length is larger than the capacity of buffer, do not need to copy the character into local buffer.
6. namespace doris
7. fill the buffer
8. retry for new content
9. requested bytes missed the local buffer
10. EOF
11. Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
12. Read
13. **comment:** Buffered Reader Add a cache layer between the caller and the file reader to reduce the times of calls to the read function to speed up.
    label: code-design
14. If the reader need the file size, set it when construct FileReader. There is no other way to set the file size.
15. Seek to the end of the file
16. Seek to a wrong position
17. Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
18. end namespace doris
19. Seek to the beginning of the file
20. buffered_reader_test_file.txt 45 bytes
21. if requested length is larger than the capacity of buffer, do not need to copy the character into local buffer.
22. buffered_reader_test_file 950 bytes

git_commits:

1. **summary:** [optimize] Optimize spark load/broker load reading parquet format file (#3878)
   message: [optimize] Optimize spark load/broker load reading parquet format file (#3878) Add BufferedReader for reading parquet file via broker

github_issues:

1. **title:** [optimize] Optimize spark load/broker load reading parquet format file

body: Currently, broker load support reading parquet file from remote, and soon we will use parquet file as be's loading source in spark load. But due to the seperated metadata (file meta/column meta/page header...) location of parquet file, broker reader need frequently seek to get data, which leads to a lot of RPCs. And large amount of RPCs will lead to huge network costs in cross-data-center scene. You can see a big gap of cost in the parquet reading below between non-cross-data-center and cross-data-center scene. (The rpc times means how many times per broker seeks during loading) **spark load** |cross-center|rpc times|load time|data size| |----|----|----|----| |No|15014|60s|560m| |Yes|16817|2h|560m| |No|169766|8min|5.8G| |Yes|150476|14h|5.8G| --- **broker load** |cross-center|rpc times|load time|data size| |----|----|----|----| |No|51780|2min|250m| |Yes|51413|42min|250m| As a proposal, **I suggest to add a cache buffer array in broker reader reading process.** **Illustration:** When a broker about to seek for a position and get data from remote parquet file, try reading with this position in the cache buffer array. Once the expected data hits the cache buffer array, then we don't bother to read data from remote parquet file. Our final purpose is to reduce the number of rpc times as much as we could, so I make some testing to validate. **Test:** The test data I used is ssb lineorder, the target table is unpartitioned with 8 buckets. | Field | Type | Null | Key | Default | Extra | |----|----|----|----|----|----| | lo_orderkey | BIGINT | Yes | true | N/A | | | lo_linenumber | BIGINT | Yes | true | N/A | | | lo_custkey | INT | Yes | true | N/A | | | lo_partkey | INT | Yes | true | N/A | | | lo_suppkey | INT | Yes | true | N/A | | | lo_orderdate | INT | Yes | true | N/A | | | lo_orderpriotity | VARCHAR(16) | Yes | false | N/A | REPLACE | | lo_shippriotity | INT | Yes | false | N/A | SUM | | lo_quantity | BIGINT | Yes | false | N/A | SUM | | lo_extendedprice | BIGINT | Yes | false | N/A | SUM | | lo_ordtotalprice | BIGINT | Yes | false | N/A | SUM | | lo_discount | BIGINT | Yes | false | N/A | SUM | | lo_revenue | BIGINT | Yes | false | N/A | SUM | | lo_supplycost | BIGINT | Yes | false | N/A | SUM | | lo_tax | BIGINT | Yes | false | N/A | SUM | | lo_commitdate | BIGINT | Yes | false | N/A | SUM | | lo_shipmode | VARCHAR(11) | Yes | false | N/A | REPLACE | --- **spark load** |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |Yes|16817|2h|560m|N/A| |Yes|613|3min|560m|128k| |Yes|529|3min|560m|512k| |Yes|285|3min|560m|1m| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |No|15014|60s|560m|N/A| |No|634|33s|560m|128k| |No|526|29s|560m|512k| |No|423|27s|560m|1m| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |Yes|150476|14h|5.8g|N/A| |Yes|6547|45min|5.8g|128k| |Yes|5670|37min|5.8g|512k| |Yes|5598|41min|5.8g|1m| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |No|169766|8min|5.8g|N/A| |No|6773|5min|5.8g|128k| |No|5667|4min|5.8g|512k| |No|5879|5min|5.8g|1m| --- **broker load** |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |Yes|51413|42min|250m|N/A| |Yes|1873|20min|250m|512k| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |No|51780|2min|250m|N/A| |No|1798|55s|250m|512k| It can be seen that whether it is cross-data-center or not, adding the cache buffer array can reduce the number of RPC and improve the loading performance. May be there still improve space in the Cache Policy, such as LRU-cache or someting else, we should try that gradually later.
label: code-design

2. **title:** [optimize] Optimize spark load/broker load reading parquet format file
body: Currently, broker load support reading parquet file from remote, and soon we will use parquet file as be's loading source in spark load. But due to the seperated metadata (file meta/column meta/page header...) location of parquet file, broker reader need frequently seek to get data, which leads to a lot of RPCs. And large amount of RPCs will lead to huge network costs in cross-data-center scene. You can see a big gap of cost in the parquet reading below between non-cross-data-center and cross-data-center scene. (The rpc times means how many times per broker seeks during loading) **spark load** |cross-center|rpc times|load time|data size| |----|----|----|----| |No|15014|60s|560m| |Yes|16817|2h|560m| |No|169766|8min|5.8G| |Yes|150476|14h|5.8G| --- **broker load** |cross-center|rpc times|load time|data size| |----|----|----|----| |No|51780|2min|250m| |Yes|51413|42min|250m| As a proposal, **I suggest to add a cache buffer array in broker reader reading process.** **Illustration:** When a broker about to seek for a position and get data from remote parquet file, try reading with this position in the cache buffer array. Once the expected data hits the cache buffer array, then we don't bother to read data from remote parquet file. Our final purpose is to reduce the number of rpc times as much as we could, so I make some testing to validate. **Test:** The test data I used is ssb lineorder, the target table is unpartitioned with 8 buckets. | Field | Type | Null | Key | Default | Extra | |----|----|----|----|----|----| | lo_orderkey | BIGINT | Yes | true | N/A | | | lo_linenumber | BIGINT | Yes | true | N/A | | | lo_custkey | INT | Yes | true | N/A | | | lo_partkey | INT | Yes | true | N/A | | | lo_suppkey | INT | Yes | true | N/A | | | lo_orderdate | INT | Yes | true | N/A | | | lo_orderpriotity | VARCHAR(16) | Yes | false | N/A | REPLACE | | lo_shippriotity | INT | Yes | false | N/A | SUM | | lo_quantity | BIGINT | Yes | false | N/A | SUM | | lo_extendedprice | BIGINT | Yes | false | N/A | SUM | | lo_ordtotalprice | BIGINT | Yes | false | N/A | SUM | | lo_discount | BIGINT | Yes | false | N/A | SUM | | lo_revenue | BIGINT | Yes | false | N/A | SUM | | lo_supplycost | BIGINT | Yes | false | N/A |

SUM | | lo_tax | BIGINT | Yes | false | N/A | SUM | | lo_commitdate | BIGINT | Yes | false | N/A | SUM | | lo_shipmode | VARCHAR(11) | Yes | false | N/A | REPLACE | --- **spark load** |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |Yes|16817|2h|560m|N/A| |Yes|613|3min|560m|128k| |Yes|529|3min|560m|512k| |Yes|285|3min|560m|1m| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |No|15014|60s|560m|N/A| |No|634|33s|560m|128k| |No|526|29s|560m|512k| |No|423|27s|560m|1m| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |Yes|150476|14h|5.8g|N/A| |Yes|6547|45min|5.8g|128k| |Yes|5670|37min|5.8g|512k| |Yes|5598|41min|5.8g|1m| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |No|169766|8min|5.8g|N/A| |No|6773|5min|5.8g|128k| |No|5667|4min|5.8g|512k| |No|5879|5min|5.8g|1m| --- **broker load** |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |Yes|51413|42min|250m|N/A| |Yes|1873|20min|250m|512k| |cross-center|rpc times|load time|data size|buffer size| |----|----|----|----|----| |No|51780|2min|250m|N/A| |No|1798|55s|250m|512k| It can be seen that whether it is cross-data-center or not, adding the cache buffer array can reduce the number of RPC and improve the loading performance. May be there still improve space in the Cache Policy, such as LRU-cache or someting else, we should try that gradually later.

github_issues_comments:

github_pulls:

1. **title:** [optimize] Optimize spark load/broker load reading parquet format file
   body: Please see the description in #3877 . spark load #3433

github_pulls_comments:

1. Great job
2. **body:** @xy720 Hi, Please add the UT for new BufferedReader class, Thanks.
   label: code-design

github_pulls_reviews:

1. Please add the comment to describe why do we need this class, and when should we use it.
2. Let the caller print the log, not here.
3. ```suggestion // If the reader need the file size, set it when construct FileReader. ```
4. ```suggestion Status _fill(); ```
5. ```suggestion Status _read_once(int64_t position, int64_t nbytes, int64_t* bytes_read, void* out); ```
6. Is it necessary to fill() when open()?
7. What is this retry for? add comment for it.
8. when will `position > _buffer_limit`?
9. Yes. If not call fill() in open(), the actual length of the buffer will be 0 when user call read().
10. ok
11. done
12. done
13. done
14. ok
15. **body:** You need to check the result, or the unit test is meaningless
    label: test
16. check the `buf_len` after `read()`
17. should it be `nbytes > _buffer_size ` ?
18. You are right.
19. what is "readat ?"
20. > what is "readat ?" Like `pread`
21. **body:** Yes, it's confusing. Better named it as pread request.
    label: code-design
22. **body:** maybe this construct funciton can be remove, for this is duplicated code. you can add default paremerts to constructor function bellow.
    label: code-design
23. good

jira_issues:

jira_issues_comments: