

git_comments:

1. * Copyright 2019 Graz University of Technology * * Licensed 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. *
2. check for empty file
3. check non-existing file
4. * Copyright 2019 Graz University of Technology * * Licensed 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. *
5. prepare file access
6. check existence and non-empty file
7. **comment:** TODO reuse blocks
label: code-design
8. core read
9. * Copyright 2019 Graz University of Technology * * Licensed 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. *
10. * Copyright 2019 Graz University of Technology * * Licensed 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. *
11. check existence and non-empty file
12. prepare file access
13. allocate output matrix block
14. * Copyright 2019 Graz University of Technology * * Licensed 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. *
15. * Copyright 2019 Graz University of Technology * * Licensed 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. *
16. copy submatrix to block
17. prepare file access
18. **comment:** TODO replication
label: code-design
19. if the file already exists on HDFS, remove it.
20. bound check
21. core write sequential
22. set up preferred custom serialization framework for binary block format

23. * Copyright 2019 Graz University of Technology * * Licensed 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. *
24. * Copyright 2019 Graz University of Technology * * Licensed 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. *
25. prepare file access
26. **comment:** for obj reuse and preventing repeated buffer re-allocations
label: code-design
27. validity check matrix dimensions
28. handle empty result
29. core write
30. if the file already exists on HDFS, remove it.
31. * Copyright 2019 Graz University of Technology * * Licensed 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. *
32. * Copyright 2019 Graz University of Technology * * Licensed 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. *
33. * * Aggregate a unary operation on this tensor. * @param op the operation to apply * @param result the result tensor * @return the result tensor
34. * * Contains the column of `DataTensor` an `_colldata` (column) index corresponds to.
35. * * Contains the (column) index in `_colldata` for a certain column of the `DataTensor`. Which `_colldata` to use is specified by the `_schema`
36. Get the dimension information from the metadata stored within TensorObject
37. sanity check correct output
38. **comment:** TODO check correct blocking
label: requirement
39. Write the tensor to HDFS in requested format
40. read tensor and maintain meta data
41. * * Aggregate a unary operation on this tensor. * @param op the operation to apply * @param result the result tensor * @return the result tensor
42. * * Contains the column of `DataTensor` an `_colldata` (column) index corresponds to.
43. * * Contains the (column) index in `_colldata` for a certain column of the `DataTensor`. Which `_colldata` to use is specified by the `_schema`
44. nnz
45. step 2: read block type step 3: if tensor allocated read its data
46. write schema and colIndexes
47. nnz, BlockType
48. read schema and colIndexes
49. `getExactSerializedSize()`, `write(DataOutput)` and `readFields(DataInput)` have to match in their serialized form definition
50. **comment:** FIXME readUTF is not supported for CacheDataInput
label: code-design
51. **comment:** TODO perf
label: code-design
52. **comment:** TODO perf, do not fill, instead save schema

- label:** code-design
53. step 1: read header (_basic, dims length, dims)
54. step 2: write block type step 3: if tensor allocated write its data
55. header size (_basic, _dims.length + _dims[*], type)
56. step 1: write header (_basic, dims length, dims)
57. * * Calculates the next index array. Note that if the given index array was the last element, the next index will * be the first one. * * @param ix the index array which will be incremented to the next index array
58. **comment:** TODO data tensor
- label:** code-design
59. **comment:** TODO choose correct value type for tensor
- label:** code-design
60. **comment:** TODO move block size to `ConfigurationManager`
- label:** code-design

git_commits:

1. **summary:** [SYSTEMDS-28] New text-cell/binary-block tensor readers/writers (local)
message: [SYSTEMDS-28] New text-cell/binary-block tensor readers/writers (local) Closes #35.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
2. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
3. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
4. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
5. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
6. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
7. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
8. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
9. **title:** Cleaning up unnecessary zero-based array initialization from LibMatrixAgg.
body: Java guarantees that new arrays will be initialized to zero, therefore these additional initializations are not necessary. cc @mboehm7
label: code-design

github_pulls_comments:

1. @deroneriksson Can you launch a test build of this for me?
2. @dusenberrymw Sure, will do.
3. Build started [here](https://sparktc.ibmcloud.com/jenkins/job/SystemML-OnDemand/57/)
4. Thanks!

5. LGTM
6. Test build passed.
7. Merged.

github_pulls_reviews:

jira_issues:

1. **summary:** Public JIRA server for tracking SystemML issues
description: Expose a JIRA server on the open Internet to track work on open-source SystemML. There are several possible options: * Install JIRA on an IBM-owned server with external visibility * Rent hosted JIRA services from Atlassian * Get access to Apache JIRA server by entering Apache incubation * Databricks creates SystemML component in Spark JIRA.
2. **summary:** Public JIRA server for tracking SystemML issues
description: Expose a JIRA server on the open Internet to track work on open-source SystemML. There are several possible options: * Install JIRA on an IBM-owned server with external visibility * Rent hosted JIRA services from Atlassian * Get access to Apache JIRA server by entering Apache incubation * Databricks creates SystemML component in Spark JIRA.
3. **summary:** Public JIRA server for tracking SystemML issues
description: Expose a JIRA server on the open Internet to track work on open-source SystemML. There are several possible options: * Install JIRA on an IBM-owned server with external visibility * Rent hosted JIRA services from Atlassian * Get access to Apache JIRA server by entering Apache incubation * Databricks creates SystemML component in Spark JIRA.
4. **summary:** Public JIRA server for tracking SystemML issues
description: Expose a JIRA server on the open Internet to track work on open-source SystemML. There are several possible options: * Install JIRA on an IBM-owned server with external visibility * Rent hosted JIRA services from Atlassian * Get access to Apache JIRA server by entering Apache incubation * Databricks creates SystemML component in Spark JIRA.
5. **summary:** Public JIRA server for tracking SystemML issues
description: Expose a JIRA server on the open Internet to track work on open-source SystemML. There are several possible options: * Install JIRA on an IBM-owned server with external visibility * Rent hosted JIRA services from Atlassian * Get access to Apache JIRA server by entering Apache incubation * Databricks creates SystemML component in Spark JIRA.

jira_issues_comments:

1. The fact that this task exists indicates that the task is complete...
2. Fred, this Task was for a public system to track SystemML work after SystemML code being made publically available on GitHub. I don't think that exists yet.
3. My mistake. Reopening this task and adding it back to the backlog.