**How to run & time results**

1. Data Generation (just plain java code, without hadoop):

java -jar hadoop-sort.jar gen delimiter , out out.txt columns first\_names.csv,last\_names.csv mb 10000

Details:

1. "java -jar hadoop-sort.jar gen" – run application with command = 'gen' (data generation)
2. "delimiter ," - result file will have ',' as delimiter between columns
3. "out out.txt" – result file name = 'out.txt'
4. "columns first\_names.csv,last\_names.csv" – each row will have 3 columns: 1) random string 20 characters (see DataGenerator.createRandomKey()); 2) random value from first\_names.csv; 3) random value from last\_names.csv. You can set up others files if you want.
5. "mb 10000" – size of result file in MB (1048576 = 1024\*1024 bytes). 10000 = 10GB as task requires

Log for 10GB data generation: logs-generate-10gb.txt

Time: 611 seconds

2. Data Sort (hadoop code):

hadoop jar hadoop-sort.jar sort in /project/in-10gb.txt out /project/out-10gb.txt

Details:

1. "hadoop jar hadoop-sort.jar sort" – run application with command = 'sort' (data sort)
2. "in /project/in-10gb.txt" – input file in hadoop catalog
3. "out /project/out-10gb.txt" – output file (sorted) in hadoop catalog

Log for 10GB data sort: logs-sort-10gb.txt

Time: 7881 seconds

3. Data Vaidation (hadoop code):

hadoop jar hadoop-sort.jar validate in /project/out-10gb.txt out /project/val-10gb.txt

Details:

1. "hadoop jar hadoop-sort.jar validate" – run application with command = 'validate' (data validation)
2. "in /project/out-10gb.txt" - input file in hadoop catalog
3. "out /project/val-10gb.txt" - output file (validation result) in hadoop catalog. If sorted file is OK, that this file will be empty.

If sorted file has error, than you will see error like "java.io.InterruptedIOException: Not valid values order" (see exampe in logs-validate-if-error.txt)

Log for 10GB data validation: logs-validate-10gb.txt

Time: 229 seconds

**Short description for java code**

Classes:

|  |  |
| --- | --- |
| **Name** | **Description** |
| StartPoint | Start point of the application. It parses arguments and run commands (gen, sort, validate) |
| DataGenerator | It generates data. Each row will have first column as ramdom text (20 characters) plus user can set up a few files as values for other columns.  Rows examples:  YVYLNLEATUWFAFMJQCTK,Nana,Luecke  LJMDWEBKDDGEUXVXSUMT,Alyssa,Shoemake  AKCIFUEDQYAGDCRMCDZW,Shawnee,Comrey |
| DataValidator | It validates data file. If sorted file has error, than user will see error like "java.io.InterruptedIOException: Not valid values order" |
| DataSorter | It sorts data with Map/Reduce. First column is a key for sorting. |