Deep Bodra
5801 1841

# REPORT

## GROUP MEMBER NAMES

Individual Project

## RUNNING THE CODE

**Run Test (test.cc)**

```
make test.out
./test.out
```

**Run gtest (bigq_test.cc)**

```
make gtest_bigq.out
./gtest_bigq.out
```

## FUNCTION DESCRIPTION

1. TwoPassMultiwayMergeSort
    a. It is a `pthread` function called by the constructor of the `BigQ`.
    b. To pass objects like `input pipe, output pipe, order maker, run length and input file` to this thread I have created a wrapper class `BigQMemberHolder`. So, these objects can be easily passed using just one parameter
    c. Phase 1 (Split Phase)
        i. The 1st phase of TPMMS involves splitting all the records into chunks called runs with each run having a `runLength` number of pages. The file containing these chunks is called a `run file`.
        ii. To track the number of pages for a run I have used a `temporary page`. I also make a copy of the current record. I try adding the record to the `temporary page`. If it fits then we are good to go. And then I add the copy of the record to the `run vector` (in-memory vector of records).
        iii. If the record doesn't fit, then I check if the `runLength` pages have been processed. If not, then I empty out the `temporary page`, add the record to the `temporary page` and the `run vector`
        iv. If the `runLength` pages have been processed then this means that `run vector` has all the records we need for this run. We use a `stable_sort`(to maintain relative order of the records) algorithm to sort the vector. The class `RecordComparator` has been added which acts as the comparator for the records.
        v. The sorted run is written out to the run file. At the same time, I track the `start page number` and `end page number` for all the runs in the vector of `RunRecordMetaData`.
    d. Phase 2 (Merge Phase)
        i. I have used a `priority_queue` to merge the records.
        ii. The head records of each of the runs is pushed into the queue.
        iii. To track the `run number` of the records I have used an object of `RunRecord` which has `record` and `runNumber` as the data members.
        iv. `RunRecordComparator` has been used to compare the records in the queue
        v. We keep removing the top element in the queue and adding the next record in that run (if any)
        vi. The removed record is inserted in the `output pipe`
        vii. We continue removing and adding records from/in the queue until the queue gets empty

## TEST CASES

TPCH data size=1GB

output1.txt

```
n_nationkey: [19], n_name: [ROMANIA], n_regionkey: [3], n_comment: [ular asymptotes are about the furious multipliers. express dependencies nag above the ironically ironic account]
n_nationkey: [22], n_name: [RUSSIA], n_regionkey: [3], n_comment: [ requests against the platelets use never according to the quickly regular pint]
n_nationkey: [23], n_name: [UNITED KINGDOM], n_regionkey: [3], n_comment: [eans boost carefully special requests. accounts are. carefull]
n_nationkey: [4], n_name: [EGYPT], n_regionkey: [4], n_comment: [y above the carefully unusual theodolites. final dugouts are quickly across the furiously regular d]
n_nationkey: [10], n_name: [IRAN], n_regionkey: [4], n_comment: [efully alongside of the slyly final dependencies. ]
n_nationkey: [11], n_name: [IRAQ], n_regionkey: [4], n_comment: [nic deposits boost atop the quickly final requests? quickly regula]
n_nationkey: [13], n_name: [JORDAN], n_regionkey: [4], n_comment: [ic deposits are blithely about the carefully regular pa]
n_nationkey: [20], n_name: [SAUDI ARABIA], n_regionkey: [4], n_comment: [ts. silent requests haggle. closely express packages sleep across the blithely]
 consumer: removed 25 recs from the pipe
 producer: inserted 25 recs into the pipe
**********************************************************************************************************************************************************************
***
 producer: opened DBFile customer.bin
 producer: inserted 150000 recs into the pipe
 consumer: removed 150000 recs from the pipe
**********************************************************************************************************************************************************************
***
o_orderkey: [4515876], o_custkey: [100685], o_orderstatus: [F], o_totalprice: [510062], o_orderdate: [1993-11-02], o_orderpriority: [4-NOT SPECIFIED], o_clerk: [Clerk#000000105],
o_shippriority: [0], o_comment: [carefully accounts: slyly ironic pinto beans detect. brave somas a]
o_orderkey: [2185667], o_custkey: [51796], o_orderstatus: [F], o_totalprice: [511360], o_orderdate: [1992-10-08], o_orderpriority: [1-URGENT], o_clerk: [Clerk#000000574],
o_shippriority: [0], o_comment: [. deposits wake quickly unusual deposits. express dependencies wake. ]
o_orderkey: [2199712], o_custkey: [66790], o_orderstatus: [O], o_totalprice: [515532], o_orderdate: [1996-09-30], o_orderpriority: [2-HIGH], o_clerk: [Clerk#000000650],
o_shippriority: [0], o_comment: [ the final, ironic deposits inte]
o_orderkey: [3586919], o_custkey: [24049], o_orderstatus: [F], o_totalprice: [522644], o_orderdate: [1992-11-07], o_orderpriority: [1-URGENT], o_clerk: [Clerk#000000924],
o_shippriority: [0], o_comment: [are alongside of the pending deposits. quick]
o_orderkey: [2232932], o_custkey: [13940], o_orderstatus: [O], o_totalprice: [522721], o_orderdate: [1997-04-13], o_orderpriority: [2-HIGH], o_clerk: [Clerk#000000245],
o_shippriority: [0], o_comment: [lithely stealthy accounts are slyly against the pend]
o_orderkey: [4576548], o_custkey: [108931], o_orderstatus: [O], o_totalprice: [525591], o_orderdate: [1997-12-26], o_orderpriority: [1-URGENT], o_clerk: [Clerk#000000336],
o_shippriority: [0], o_comment: [he slyly ironic requests. regular, bold deposits cajol]
o_orderkey: [3043270], o_custkey: [144617], o_orderstatus: [O], o_totalprice: [530604], o_orderdate: [1997-02-12], o_orderpriority: [5-LOW], o_clerk: [Clerk#000000699],
o_shippriority: [0], o_comment: [riously final deposits? ]
o_orderkey: [4722021], o_custkey: [128120], o_orderstatus: [F], o_totalprice: [544089], o_orderdate: [1994-04-07], o_orderpriority: [1-URGENT], o_clerk: [Clerk#000000230],
o_shippriority: [0], o_comment: [al, express pinto beans are after the careful]
o_orderkey: [1750466], o_custkey: [21433], o_orderstatus: [F], o_totalprice: [555285], o_orderdate: [1992-11-30], o_orderpriority: [4-NOT SPECIFIED], o_clerk: [Clerk#000000040],
o_shippriority: [0], o_comment: [ ironic packages. even notornis integrate. fluffily regular foxes snooze furi]
 consumer: removed 1500000 recs from the pipe
**********************************************************************************************************************************************************************
***
n_nationkey: [7], n_name: [GERMANY], n_regionkey: [3], n_comment: [l platelets. regular accounts x-ray: unusual, regular acco]
n_nationkey: [19], n_name: [ROMANIA], n_regionkey: [3], n_comment: [ular asymptotes are about the furious multipliers. express dependencies nag above the ironically ironic account]
n_nationkey: [22], n_name: [RUSSIA], n_regionkey: [3], n_comment: [ requests against the platelets use never according to the quickly regular pint]
n_nationkey: [23], n_name: [UNITED KINGDOM], n_regionkey: [3], n_comment: [eans boost carefully special requests. accounts are. carefull]
n_nationkey: [4], n_name: [EGYPT], n_regionkey: [4], n_comment: [y above the carefully unusual theodolites. final dugouts are quickly across the furiously regular d]
n_nationkey: [10], n_name: [IRAN], n_regionkey: [4], n_comment: [efully alongside of the slyly final dependencies. ]
n_nationkey: [11], n_name: [IRAQ], n_regionkey: [4], n_comment: [nic deposits boost atop the quickly final requests? quickly regula]
n_nationkey: [13], n_name: [JORDAN], n_regionkey: [4], n_comment: [ic deposits are blithely about the carefully regular pa]
n_nationkey: [20], n_name: [SAUDI ARABIA], n_regionkey: [4], n_comment: [ts. silent requests haggle. closely express packages sleep across the blithely]
 consumer: removed 25 recs from the pipe

```

## GTEST

The thread `TwoPassMultiwayMergeSort` generates an intermediate file called the *run file* in the first phase. This file is named “yolo.runfile” in this project. The thread also generates a *bigq* file which contains the records in the sorted order. I have written 4 gtests which tests these files for various aspects.

**Notes before executing gtests**:

1. These tests are to be run after the `test.cc` has been executed with `3. Sort and Write option`
2. CNF for gtest must be same as you used when you executed `test.cc`

**Tests**

1. RunFileExistenceTest
    a. It tests whether the run file by the name “yolo.runfile” was created or not
2. RunFileSizeTest
    a. It tests whether the file size of the *runfile* is as at least as that of the input *bin file* or not
3. NumberOfRecordsTest
    a. It tests whether the number of records in the output *bigq file* is equal to that in the input *bin file* or nor
4. SortOrderTest
    a. It tests whether the number of records that are out of order in the *bigq file* is 0 or not

**Result**

**1GB Data**

Test Case

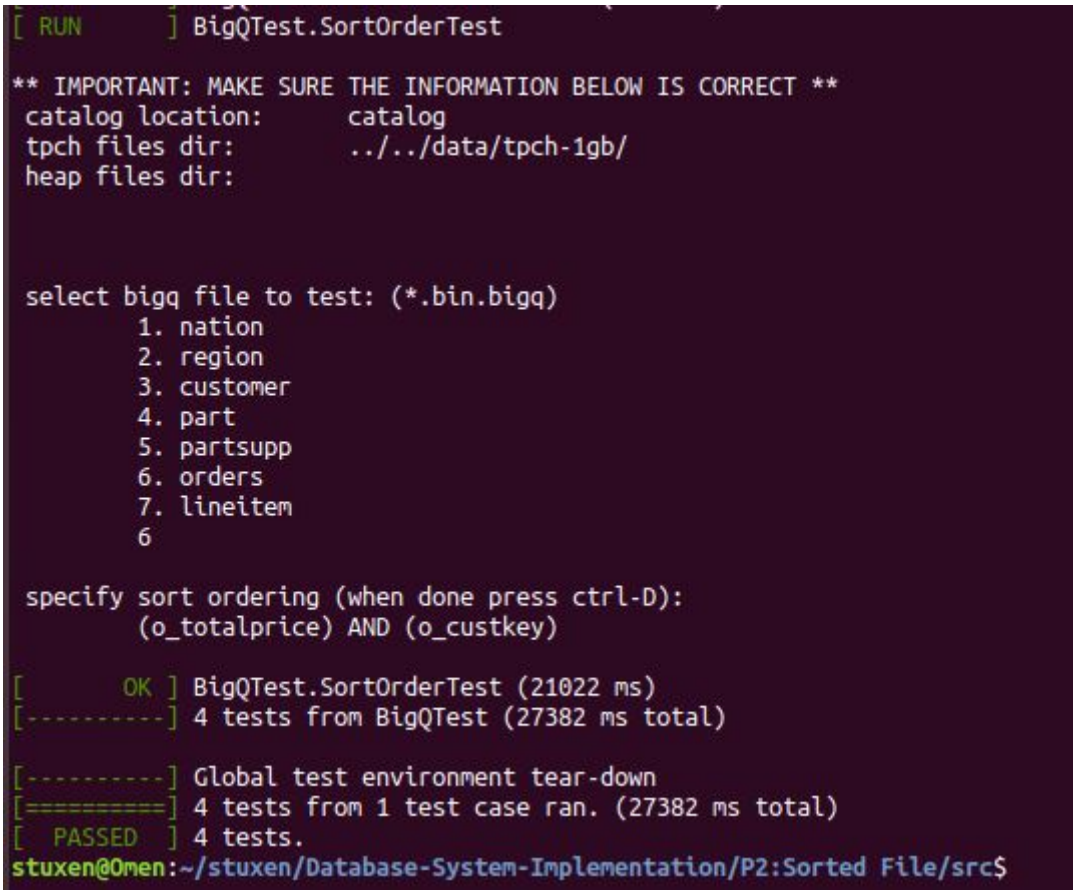
```
Test Index: 3. Sort and Write
Find Index: 6. orders
Run length: 128
CNF: (o_totalprice) AND (o_custkey)
```

1. ./test.out

```
stuxen@Omen:~/stuxen/Database-System-Implementation/P2:Sorted File/src$ ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        ../../data/tpch-10mb/
 heap files dir:


 select test option:
        1. sort
        2. sort + display
        3. sort + write
        3

 select dbfile to use:
        1. nation
        2. region
        3. customer
        4. part
        5. partsupp
        6. orders
        7. lineitem
        6

 specify runlength:
        128

 specify sort ordering (when done press ctrl-D):
        (o_totalprice) AND (o_custkey)

 producer: opened DBFile orders.bin
 producer: 100000
 producer: 200000
 producer: 300000
 producer: 400000
 producer: 500000
 producer: 600000
 producer: 700000
 producer: 800000
 producer: 900000
 producer: 1000000
 producer: 1100000
 producer: 1200000
 producer: 1300000
 producer: 1400000
 producer: 1500000
 producer: inserted 1500000 recs into the pipe
 consumer: removed 1500000 recs from the pipe
 consumer: recs removed written out as heap DBFile at orders.bin.bigq
 consumer: 1500000 recs out of 1500000 recs in sorted order
stuxen@Omen:~/stuxen/Database-System-Implementation/P2:Sorted File/src$
```

2. `./gtest_bigq.out`

```
stuxen@Omen:~/stuxen/Database-System-Implementation/P2:Sorted File/src$ ./gtest_bigq.out
[==========] Running 4 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 4 tests from BigQTest
[ RUN      ] BigQTest.RunFileExistenceTest
[       OK ] BigQTest.RunFileExistenceTest (0 ms)
[ RUN      ] BigQTest.RunFileSizeTest

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        ../../data/tpch-1gb/
 heap files dir:



 select bigq file to test: (*.bin.bigq)
         1. nation
         2. region
         3. customer
         4. part
         5. partsupp
         6. orders
         7. lineitem
         6
[       OK ] BigQTest.RunFileSizeTest (1714 ms)
[ RUN      ] BigQTest.NumberOfRecordsTest

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        ../../data/tpch-1gb/
 heap files dir:



 select bigq file to test: (*.bin.bigq)
         1. nation
         2. region
         3. customer
         4. part
         5. partsupp
         6. orders
         7. lineitem
         6
[       OK ] BigQTest.NumberOfRecordsTest (4646 ms)
[ RUN      ] BigQTest.SortOrderTest
```


[ RUN ] BigQTest.SortOrderTest
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ../../data/tpch-1gb/
heap files dir:
select bigq file to test: (*.bin.bigq)
1. nation
2. region
3. customer
4. part
5. partsupp
6. orders
7. lineitem
6
specify sort ordering (when done press ctrl-D):
(o_totalprice) AND (o_custkey)
[ OK ] BigQTest.SortOrderTest (21022 ms)
[----------] 4 tests from BigQTest (27382 ms total)
[----------] Global test environment tear-down
[==========] 4 tests from 1 test case ran. (27382 ms total)
[ PASSED ] 4 tests.
stuxen@Omen:~/stuxen/Database-System-Implementation/P2:Sorted File/src$

## CONCLUSION

We have taken a stepping stone towards implementing the sorted file