COP6726 Report of Project2-Part1

Group member:

Jiajing Liao(UFID: 01469951)
Haocheng Song(UFID: 11851321)

Changing for source code:

I think source code for OrderMaker doesn't work, because original OrderMaker would verity both sides of CNF like (r_name = "ASIA"). But in this project, we would only get input from left side of CNF like (r_name) without right side. So if we continuing using original OrderMaker, it will judge CNF not valid because it lack for right side, and OrderMaker would not act like a sorting order.

Me and other students tried multiple times and it turns out using original OrderMaker would not make record actual sorted. Even command line would show all records sorted, but that is because test.cc also used OrderMaker to determine whether records are sorted.

So I change OrderMaker a little bit to make it no longer verify the right side of CNF. I made line 146 commented so that OrderMaker would no longer verity right side of CNF.

```
141         }
142
143         // now verify that it operates over atts from both tables
144         if (!((orList[i][0].operand1 == Left && orList[i][0].operand2 == Right) ||
145               (orList[i][0].operand2 == Left && orList[i][0].operand1 == Right))) {
146             //continue;
147         }
148
149         // since we are here, we have found a join attribute!!!
150         // so all we need to do is add the new comparison info into the
151         // relevant structures
```

After changing, all records become actual sorted after BigQ sorting.

Compiling and running project:

Please make sure you are using linux system, this makefile don't work for mac system, contact haochengsong@ufl.edu if you need makefile for other operating system.

Steps of compiling and running test.out(after entering directory of project):

1. make clean
2. make
3. make test.out

4. ./test.out

Step of running test cases script(result will be output1.txt):
1. make clean
2. make
3. ./runTestCases.sh

The directory for storing .bin file is hardcoded in test.h file. Please make change in test.h to match your .bin file.

For compiling and running gtest, refer to the end of this report.

Explanation for classes and methods:

Class Run:

This class represent the "Run" which consist of sequence of pages. In the phase one, records will be gathered into a Run, and then do the phase to do merge-sorting for runs, to output sorted records. Noticed that Run is just a abstract of paged stored in file, so file is the actual disk base for storing records when we do sortin. Below are methods or important variables of Run.

Run::Run(File* file, int startPage, int runLength);

Construct method for run, it will take a file pointer and a startPage index which indicate the start page for run, and also the length of run(how many pages it stored).

Record *topRecord;

This variable is the top record stored in Run. Since run is sorted in given order, so topRecord is the current smallest element in the Run.

int Run::UpdateTopRecord();

This method is used to pop the current top record, and get next record to be the top record from Run. Return 0 if there are no more records in this Run.

Class RecordComparer:

This class is customer comparator for records. Basically it will compare records using given OrderMaker.

RecordComparer::RecordComparer(OrderMaker* orderMaker);

Construct method for RecordComparer, it takes OrderMaker as a parameter, which is used for comparing.

bool RecordComparer::operator () (Record* left, Record* right);

This is customer comparing operator, it will do comparing base on give OrderMaker.

Class RunComparer:

This class is customer comparator for runs. Basically it will compare runs using given OrderMaker.

RunComparer::RunComparer(OrderMaker* orderMaker);

Construct method for RunComparer, it takes OrderMaker as a parameter, which is used for comparing.

bool RunComparer::operator () (Run* left, Run* right);

This is customer comparing operator, it will take top record from each runs, and compare top records to decide order of runs.

Struct WorkerArg:

This struct is used for transferring parameter from BigQ to worker's functions.

Class BigQ:

This class is the base class for sorting records. It only has a construct method to handle input.

BigQ :: BigQ (Pipe &in, Pipe &out, OrderMaker &sortorder, int runlen);

Construct method for BigQ, it will build a WorkerArg which is used for worker'function. Then it will create worker thread, and let worker thread to do rest of things.

Worker thread:

Worker thread is responsible for retrieving all records from input pipe, sort them and put sorted records into output pipe. There are two functions without class that is used for worker.

void* workerMain(void* arg);

Main method executed by worker, worker will retrieve records from input pipe, sort records into runs and put all runs into priority queue, and get sorted records from priority queue to output pipe .

void* recordQueueToRun(priority_queue<Record*, vector<Record*>, RecordComparer>& recordQueue, priority_queue<Run*, vector<Run*>, RunComparer>& runQueue, File& file, Page& bufferPage, int& pageIndex);

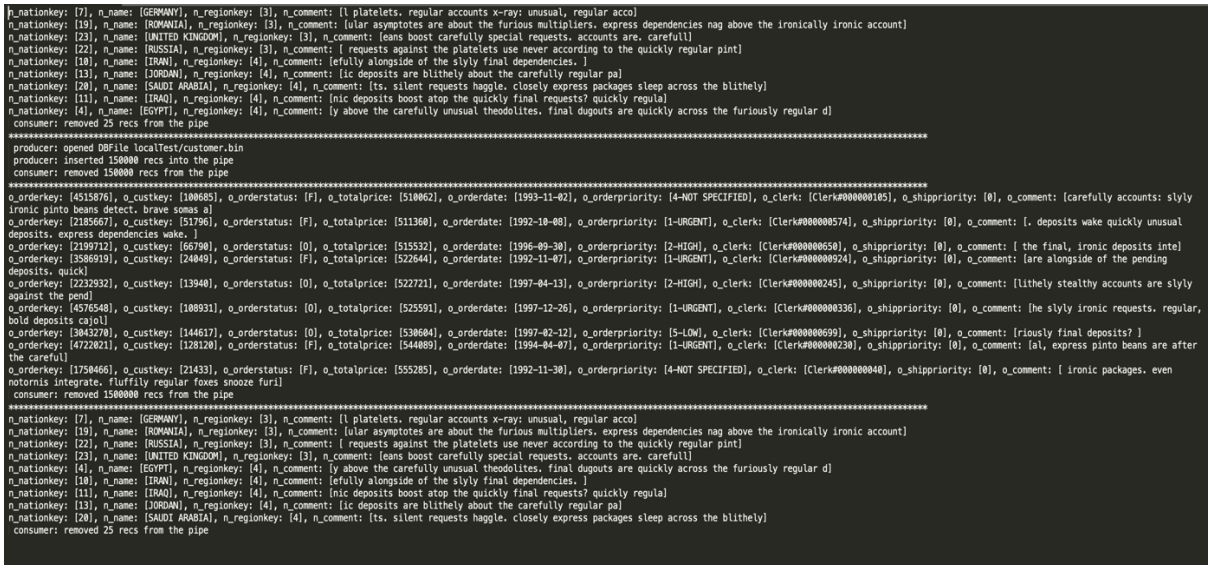
Used for take sequences of pages of records, and construct a run to hold such records, and put run into priority queue for runs.

Output1.txt results:

Below are results from runTestCases.sh

They are sorted by given order from tc1.txt, tc2.txt, tc3.txt and tc4.txt. I changed OrderMaker a little bit to make result be actual sorted, otherwise records would not be actual sorted if we

use original OrderMaker.



Gtest results:

To running the gtest, first make sure you enter under root directory of project, where should contain gtest.cpp file Makefile.
You must have gtest installed on your computer to running my gtest.
https://github.com/google/googletest/blob/master/googletest/README.md#standalone-cmake-project
Steps for running gtest:

1. make clean
2. make gtest
3. ./gtest.out

There are totally 4 tests that testing Run for updating top record; testing RecordComparer for correctly compare records; testing RunComparer for correctly compare runs; testing non-class method recordQueueToRun whether it can correctly putting records into run. Below is screen shot for gtest.