Concordia University
Dept. of Computer Science and Software Engineering
COMP 6521: Advanced Database Technology and Applications
Winter 2019-2020

Lab Assignment 1

PLEASE note the changes below in the record structure:

Total Points: 8
Codes & Reports: Wed. Feb. 19, 2020 at noon
Demos: Wed. Feb. 19, 2020

Project Description: We have the data files T1 and T2 with the same schema as follows:

EmpID INT (8)
LastUpdate CHAR (10)
EmName CHAR(25)
Gender INT(1)
Dept. INT(3)
Social Insurance Number INT(9)
Address CHAR(43)

Date is a string of length 10 characters of the form 'YYYY-MM-DD'. Each of these two files is stored as a text file, and each tuple is of size 100 bytes, stored as a separate line/record/row in the file. Each line is ended with the "return" key. Suppose each block is of size 4K, holding 40 tuples. The rest of the block is left unused. Furthermore, the blocks of each file are stored in consecutive disk blocks. There is not delimiter separating the values of different fields/attribute. An example of a tuple in these files is as follows (but would be on a single line):

123456782020-01-24John 12223333333331455 de Maisonneuve West, Montreal, QC, H3G 1M8, Canada

Each of these files could contain tuples with duplicated Employee IDs. In this application, we say that tuples t1 and t2 are "duplicated" if they have the same EmpID's. (Note that this definition is not the same as the standard one which says t1 and t2 are duplicates if they have the same values for every attribute.) In T1 and T2, a tuple could have any number of "duplicates." In this project, you and your team are required to merge these two files as follows: for any duplicates of a tuple t, the final merges file should contain only the one that was most recently updated; all earlier updates of t will not be present in the output. Implement the TPMMS method and use it to produce the desired output. For instance, sort T1 and T2 first, and then identify and eliminate the duplicates in the merge phase. Report the number of output tuples, the number of blocks holding these tuples, the total number of disk I/O's to perform the task, and the processing time.

Ignore the number of the disk I/O's and the time required to write the final result back to the disk.

You need to evaluate the performance of your implementation using large instances of T1 and T2. The lab instructors will use instances of T1 and T2 which you can use to test and evaluate the performance. Your report should include the test results, at least on these instances. You may include additional test results using the instances created by your team.

To further evaluate the performance of your implementation, consider instances of T1 and T2 at around 500,000 and 1,000,000 tuples, respectively. Study the performance of your implementation in the following two cases of restricted main memory available: (1) 10 MB and (2) 20 MB.
Run the experiments in each of these two cases and report the following:

- Compare the number of disk I/O's and the execution time, in seconds, for the sort operations given in each case of the two main memory sizes. This should include the time to write the sorted data back to the disk.

- Compare the number of disk I/O's and the execution time for performing the whole task in each case of main memory sizes.

What tools you should use?

Use VM argument Xmx5m in Eclipse to restrict the main memory sizes in Java Virtual Machine.

What to submit by the due date?

Through Moodle, submit your report (in a single PDF format) and the source codes (a single zip file). This also includes instruction to compile and run your code. Make sure your program compiles and runs on the computers in the labs H-903 and H-907 assigned to the course.

Demos: Book a time slot with the lab assistants for the demo of your project on Feb. 19th. Each member of a team must be present in their project demo.

Bonus: The lab instructors may recommend additional 2 points for the implementation with the best performance and additional 1 point for the next best.