Concordia University
Dept. of Computer Science & Software Engineering
COMP 6521: Advanced Database Technology and Applications
Winter 2023

Lab Mini-Project 1

- Maximum Points: 9
- Report submission due: Tuesday Feb. 21st through moodle
- Demos: Wednesday Feb. 22 during the Lab sessions

Project Description: In this project, you and your team are required to implement the Two-Phase, Multiway Merge-Sort Method and evaluate your program a *bag-based union*, denoted as *bbu*(R1,R2), where R1 and R2 are two compatible derived tables, i.e., each may contain multiple copies of the same record. The structure of the records is as follows:

1. Student ID:	int (08)
2. First Name:	char(10)
3. Last Name:	char(10)
4. Department:	int (03)
5. Program Code:	int (03)
6. SIN Number:	int (09)
7. Address:	char(57)

Each of these tables is originally stored as a separate file. If they were stored on disk, there would be in consecutive disk blocks. Assume the block size is 4K bytes and that each block holds 40 tuples. Viewing each table as a file, each record appears on a separate line. There is no symbol used to separate different components of a record; they are distinguished by the record offset, that is, the first 8 digits indicate the Student ID, the next 10 characters indicate the first name, followed by the last name in the next 10 characters, etc. You should use the length of the characters to identify and/or extract different component values in a record. Here is an example of a record:

```
11111111John      Smith     4445556666666661455 Maisonneuve West, Montreal, QC, H3G 1M8
```

In your implementation, develop the TPMWMS method and use it to R1 and R2 separately, and then produce the desired output in the merge phase. The output should be in the compact form, that is*, **if R1 has n1 copies of t, and R2 has n2 copies of t, then the output produced includes tuple t:n, where n=n1+n2.*** Note that the tuples in R1 and R2 and not in compressed form. That is the n1 copies of t are scattered in R1 and not necessarily appearing together. Write the blocks of output records of the form t:n to consecutive disk blocks, each containing 40 tuples of the form t:n.

Evaluate and report the performance of your program using instances of R1 and R2, which we will make available in moodle. Your report should include the following results in the form of a table:

1. Report the total number of disk I/O's and time in ms for Phase 1. To present all the processing times in your report, use the formula s+r+t, where s is the average seek time, r is the average rotational delay, and t is the time to transfer one block, and s+r+xt if transferring x number of consecutive blocks at once. Report x(s+r+t) if you are transferring x random blocks.
2. Report the total number of disk I/O's and the time for Phase 2 for producing the output in the main memory Phase 2. Do not need to report the time for writing the final result to disk.

3. Report the total number of output records, the number of output blocks written to disk, and total time to produce the output in the main memory.

You need to create large instances of R1 and R2 (each having more than 1 million records), evaluate and improve the performance of your implementation. However, in your report, consider the the instances we provide and present the results. We will evaluate your code based on use the reported results to evaluate correctness of your code and compare the work of different groups.

To study the impact of the amount of available main memory M, limit M as suggested in the following two cases and report the above 3 items:

(1) M = 51 blocks
(2) M = 101 blocks

Evaluate the performance of your implementation by executing experiments in each case of these two M values. Your report should include a comparison of the TOTAL number of disk I/O's and execution times for sorting and merging these files using only the amount M of the main memory available.

Tools to use:

Use VM argument Xmx5m in Eclipse to restrict the main memory usage of Java Virtual Machine. The lab assistants can help in using Xmx5m.

What to submit on due date:

Submit your project report and the source codes through moodle. Please include instructions to compile and run your code. Make sure your program compiles and runs on the lab computers.

Book a time slot for your project demo on Feb. 22nd. We will put up a schedule in moodle to pick a time. Both lab instructors will be present during the demos for evaluating your projects. Every member of your team MUST be present during your project demo.

Bonus: The lab instructors may recommend additional 1 or 2 points if you also study and report the scalability issue as the number of records in each of R1 and R2 increases to 500,000 and 1000,000.