

CSCI4370 Database Management

Spring 2013

Project 4: Performance Evaluation of RA Operators.

Due: March 18th (Monday), 8 am.

In this project you will plot performance for Selects and Joins (response time in msec vs. number of tuples).

Compare the following:

- Sequential select vs. indexed select.
- Nested loop join vs. indexed join.
- ArrayList vs. FileList.
- TreeMap vs. your index structure (B+tree and Hash).

You can find brief info on nested loop join here:

http://en.wikipedia.org/wiki/Nested_loop_join.

Print your index structure. Also, present performance results. Gold (+60), silver (+40), and bronze (+20) medals will be awarded for best performers.

For plotting the graph, you need to collect enough data to be able to make comparison. Then you need to draw error bars using collected data points. You can plot error bars using standard error, which is based on standard deviation. Without error bars, you can't have scientifically definitive statements. You can find a brief explanation on calculating and drawing error bars here:

http://en.wikipedia.org/wiki/Standard_error#Assumptions_and_usage.

So you need to get standard deviation and the mean, then get those in the formulas (refer to the link above.) and plot your error bars.

You can present performance results using Excel diagrams. You will prepare three charts with the following components:

1. **Select - Point Query:** $\sigma_{id=v}(Student)$
 - Table Scan – ArrayList
 - Table Scan – FileList
 - Indexed Select - FileList, TreeMap
 - Indexed Select - FileList, B+Tree

- Indexed Select - FileList, Linear or Extendible Hash
- 2. **Select - Range Query:** $\sigma_{gpa>3.0}(Student)$
 - Table Scan - ArrayList
 - Table Scan - FileList
 - Indexed Select - FileList, TreeMap
 - Indexed Select - FileList, BpTree
- 3. **Join:** $Student \text{ join }_{id = studId} Transcript$
 - Nested Loop Join - FileList
 - Indexed Join - FileList, TreeMap
 - Indexed Join - FileList, BpTree
 - Indexed Join - FileList, Linear or Extendible Hash

Reuse your implementation and source code in Projects 1, 2 & 3. Three new files are TupleGenerator.java, TupleGeneratorImpl.java, TestTupleGenerator.java (at course web page). These files may include more than what you need for this project.

Your program must be thoroughly documented (generate javadoc). Use the @author tag for each class and method. Each method should have a single author. The coding workload should be split roughly equally among four team members. We will check this by examining the @author tags. Please make sure that the output of your program is easy to understand. Provide a flag for turning on/off your tracing/debugging messages in your program's output – if necessary.

Programming language: Java 7 is required for the project.

What to submit: Please submit

- all source code
- all the javadoc files
- a readme file
- performance diagrams in Excel or PDF

The readme file should contain: your names, how to compile and run your code and other specifications you want to make. Please pack all your files in a zip package with the file name: "project4" + last names of group members. For example: project4_chen_kim_zhang_luo.zip

How to submit: Mail your ".zip" file to the TA (Ugur Kursuncu, kursuncu@uga.edu)

An electronic copy of this project description and Java source code templates to be used can be found at course web page.