Collin Gros
05-07-2021
CS-482
Phase 2

PHASE 2 PROJECT REPORT

PROBLEM

Create a UI to handle MySQL queries and inserts/deletions. Iteratively insert and bulk insert large datasets into a database using this UI.

METHODS

I used Java and took advantage of Swing libraries to create a Button/Text-Field based UI for handing MySQL queries. I used JDBC from DriverManager in Java to connect to my MySQL server.

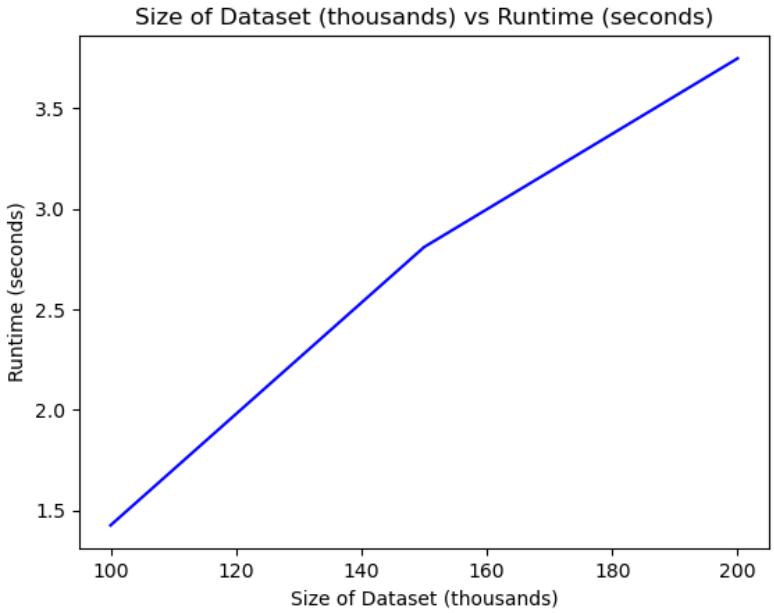
I randomly generated data in sizes of 100K, 150K, and 200K to test bulk insertion and iterative insertion. This was done in a function in my Java code.

Tests were run using an all-purpose function in my code.

I used Python and matplotlib to generate graphs for my report.

RESULTS/ANALYSIS

BULK LOADING



Bulk Loading performed with a linear scaling in respect to dataset size. As you can see, the *players.txt* file with 150K entries took the longest to execute. However, I would minimize this difference as it probably was caused from background processes running on my machine. The largest dataset (200K) took less-than-expected time to execute than the 150K dataset - which makes me think that the runtime scales linearly with the dataset size.

ITERATIVE INSERTION

Unfortunately, I do not yet have a graph available to view, as the program is still being executed on my machine. I will upload my data and graph through the Comments section on Canvas.

However, during the video demo, I saw that the runtime was about 1230 seconds, which equates to about 20 and a half minutes. Because I am still running tests, I will expect the 150K test to be 150% times as long as the first test, which would be 31 minutes long. I can also assume the 200K test to take 42 minutes, judging from the first result.

I would assume that the runtime will also scale linearly, similarly to the results gathered from Bulk Loading. In my code, I parse the data in a 2D array using a nested for-loop. I assume that the runtime will upscale linearly with data size, but I will find out.