Profiling

Branislav Dubec, [xdubec01]

20. 4. 2019

In this file there are four graphs that depict calling of functions in program profile made by “valgrind” with tool “callgrind”.

Each mathematical operation is made by our mathematical library originally made for the school project.

The main graph shows the time spent and number of calling of each function. This includes functions to generate numbers and to get the standard deviation of these numbers. For those I made three graphs that describe the calling with 10, 100, 1000 individually.

I have analyzed these graphs and tried to find out which of the functions is the best for optimizing. I have decided to make conclusion based on function “getRandom1000”, for the most accurate results.

The function “getRandom1000” randomly makes 1000 numbers, adds them together (for the formula of standard deviation); that’s why the “addition” function is called 2000 times (one that increments value in loop).

The function “getSmer” adds together these 1000 numbers to the power of 2; and calls the function “result” to calculate the formula “sqrt(1/(CountofNumbers-1)*(Sumof Numbers^2-SumofNumbers))”. Because the formula is in bracket, most of the time is spent in function “BracketEvaluation” that performs calculations inside parentheses. In my opinion, the time spent in other functions for calculations is adequate.

However, I noticed that time spent in functions for string manipulation, as the whole formula was taken as string, is pretty high. After consultation with my teammates, we found out that if we used other way for parsing the expression and minor calculations, the overall performance would have been better.