**Lab 4: Performance Analysis**

In this lab, sorting functions like in lab 3 were used via a trans compiled C++ codebase. Web sockets were utilized to connect a node.js server with the trans compiled C++ client. Basic architecture of the project is as follows; the node server sends a list of 1000 unsorted integers to the client at which point the client performs the appropriate sorting operation on the list and sends the now sorted list back to the server. 3 different timing performance analyses were done to reflect the structure of this project’s codebase: timing of server (time between server sending an unsorted list and receiving the sorted list), timing of client (timing of the sorting operations done in trans compiled client code), and finally, timing for standard native C++ execution of the algorithms. All timing data were recorded in microseconds and repeated over 20 iterations.

Immediately, it is obvious that server timing fares the worst. On average, timing of server is around twice as large as timing of client. This is logical as timing of server would be timing of client plus time it takes for web socket communication. As for the native C++ code, it unsurprisingly performs exponentially better than both client and server timing. From this overview of timing comparisons of different codebases, it can be clearly seen how increasing level of overhead from trans compilation to web socket communication add to a decrease in timing performance.

The 3 charts above take a closer look at the individual algorithms and their timing performance subjected to 3 different environments. These 3 charts also show statistical significance between the three environments as there are considerable gaps between each environment at 95% confidence interval. Even though the timing difference between native C++ and trans compiled client code is significant, their difference looks minimal when compared to timing data of server executions. It can be seen very clearly that web socket communication and passing of data back and forth take a significant toll on timing performance.