# Exploring the Performance of Server-Client Socket Communication

## Abstract

The report is going to discuss the design and implementation of a server-client number operation service model. In this model, the client sends a sequence of numbers to the server for operations such as square operation or square root operation, then the server uses multiple threads to process the number and returns results to the client. Two versions of server are involved and a series of test is conducted to explore the performance gap between them, with several plots showing relevant data. Conclusions about the experimental evaluation are also drawn in the end.

## Design

The project is divided into three parts (packages): Server part, Client part and Communication tool part.

The Server part consists of two versions of server program and a ServerThread class functioning as the threads of the server. Both servers have 10 threads. One version of server uniformly divides the number sequence into each thread, the other version uses k-1 threads for the first k-1 numbers and the last thread for all the remaining numbers.

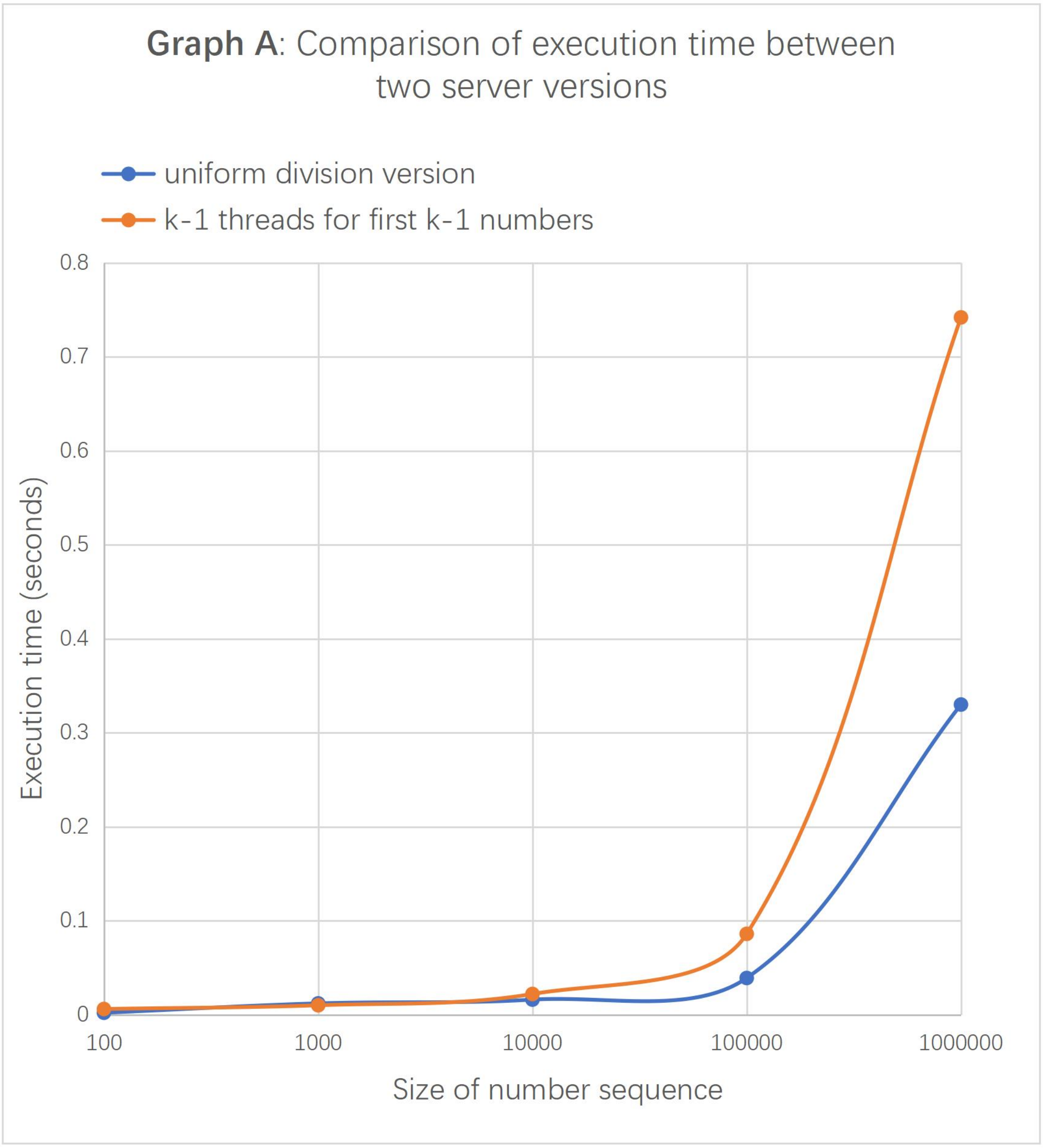
The Client part includes a client program for users to use and another client program for data testing purpose.

The Communication tool part supports the data transmission between the server and the client. It includes two classes. MessageSender class is used to generate objects transmitted between the server and the client. ServiceProtocol class focuses on processing data from the client and preparing the results.

## Testing

To test the performance of these two server versions, each server is given 10 threads and the range of each number is between 1 to 1000. Then the size of number sequence is increased to compare the execution time of each server. Since the speed of computer is very fast, the sequence size should reach a large number to enlarge the performance gap between servers. The sequence size is orderly set to a hundred, a thousand, ten thousand, a hundred thousand and a million to record the execution time.

The data plot is presented below. Every case has been run 5 times to get an average value in order for accuracy.



It is clearly represented in Graph A that as the size of number sequence goes up to a huge number, the performance gap between two server versions becomes remarkable. After the size of number sequence reaches 100,000 the gap becomes apparent and the execution time of the first version (blue line) is nearly only a half of the other (orange line). It is sensible to deduce that as the sequence size continues to become larger, the performance gap would also continue to become larger and larger.

## Conclusion

From the discussion and analysis of testing above, it could be concluded that the performance of the uniform division version of server is much better than the other version of server, which uses k-1 threads for first k-1 numbers. The performance gap between them implies that reasonably utilizing threads would greatly improve the performance of a program, which is especially important to a server program.