Computer Networks - Prime Number Project

Group M:
Devon Jarvis - 1365149
Jiahao Huo - 1475365
Kyle Rip - 1076026
Meir Rosendorff - 1490527
Or Hanoch - 1501858

May 17, 2017

Part I
Introduction

The objective of this project was to create a client-server enviroment that will recieve a range of natural numbers and return the amount of prime numbers within that set. The enviroment consists of 1 client and 3 servers.
This project has been programmed in python and uses the socket and pickle libraries.

1 Input and Outputs

Input: Lower bound of the set (int) and Upper bound of the set (int)
Output: Amount of prime numbers in the set (int)

2 Files

- TCPClient.py - Runs the client
- TCPServer1.py - Runs server no. 1
- TCPServer2.py - Runs server no. 2
- TCPServer3.py - Runs server no. 3
- primeCalc.py - Contains the methods countPrimes and checkingPrime, called from the servers.

2.1 TCPClient.py

Runs the client and performs the following:

1. Initializes three separate TCP, IPv4 sockets and sends connection request to three different servers.
2. Requests input of lower and upper bound limits for the range that will be analysed.
3. Splits the range given into three equal sections and creates arrays with specifying them.
4. Starts a timer that will count the amount of time it takes to perform calculations.
5. Using "pickle" library sends the each section array to a different server to be analysed.
6. Using "pickle" library recieve results from servers.
7. Stop timer.
8. Print output of amount of primes in each section and time it took, aswell as the total result of the entire range and time it took.

Note: We first send data to all servers and then recieve data from server (as opposed to sending and recieving to one server at a time) in order to have the calculations run in overlapping time periods on all servers, thus utilizing the maximum amount of hardware at any given time, making the complete calculation faster.

2.2 TCPServer#.py

Runs a server and performs the following:

1. Initializes TCP, IPv4 socket and accepts a connection from a client.
2. Recieves (using "pickle library") array with range of numbers to be analyzed.
3. Sends the range to countPrimes in primeCalc.py and gets response of amount of primes in range and how much time it took.
4. Sends results back to client.

2.3 primeCalc.py

This file consists of two functions: checkingPrime and countPrimes.

2.3.1 checkingPrime

Checks if a given number is a prime number.[1]
Input: int - an integer to be checked if it is a prime.
Output: boolean - a true or false value of whether the input is or isn't a prime.

2.3.2 countPrimes

Goes over a range of integers and checks how many primes are in that range and the amount of times it takes to calculate it. Uses checkingPrime to evaluate whether an integer in the range is a prime.
Input: int,int - lower bound of range, upper bound of range.
Output: int, time - amount of primes in range, amount of time calculation took.

3 Project Member Roles

Devon Jarvis - Research, Structuring, Overviewing and Writing Code
Jiahao Huo - Research, Structering, Overviewing and Writing Code
Kyle Rip - Research, Overviewing, Reformating
Meir Rosendorff - Research, Structuring, Debuging & Optimizing, Overviewing, and Documentation
Or Hanoch - Research, Structuring, Overviewing and Writing Code and Documentation

[1] Uses algorithem found in Project Euler Problem 7: https://projecteuler.net/overview=007