Homework 5

Prof. Eero Vainikko
Teaching Assistant: Tek Raj Chhetri, Faisal Ahmed
Parallel Computing
Deadline: 1 Week from the date assignment is opened

October 7, 2019

1 Objective

Apply the MPI technique learned from earlier assignment to simulate the streaming data.

2 Assignment

In today’s world data is everything. There are lots of data being generated. Those generated huge amount of data needs to be processed otherwise it means nothing. There are many techniques and one of them is stream analysis where the data are constantly sent to the intended service where analysis is made to the incoming data. One way for performing such tasks could be using Apache Spark i.e. Spark Streaming. In this assignment we are also going to simulate the same using MPI, in future we will also look into Apache Spark.

For this assignment you need to perform the task following the flow as shown in figure below. The child initiates the request to the master. The master then based on the request of the child sends some data. Each time child sends request, master response with some data i.e. some string data. The string data that master sends here is the random data generated using the script [1]
res = ''.join(random.choice(string.ascii_uppercase + string.digits)
for i in range(180))

Listing 1: Random String Generation

Once child receives the data from master it should perform the following task.

- Find the sub-string from the generated random string by the master by matching the specific pattern such that it begin and end with a number. The matched substring should be of a minimum of length 3 excluding the starting and ending number. Eg: 8RCVT9,1RCA4

- The sub-string should be unique.

- The sub-string can contain numbers but should be unique as well. Eg: 13YRW4 is allowed but 11RSG3, 1SFES33 is not allowed.

- The starting number of sub-string should be smaller than the ending one. Eg: 1REWT2 is allowed but 4FSDF2 is not allowed.

- Once the task is completed it keeps the result and just report back the number of sub-strings that matched the pattern.
Figure 1: Task