Cloud Computing

Wiktor Susfał, 276400

This project covers the topic of running some calculation program on a virtual machine hosting in Amazon AWS Cloud.

1. Description of the program

The program is aimed to calculate a matrix expression given to it by the user and has been built with the use of Python 3.5.1 programming language. The expression is a custom combination of adding and multiplication actions. The expression should contain only matrices.

Additional assumptions are:

- the matrices are stored in text files (with .txt format) in a folder called *matrices,* so there should be the .py file with code and *matrices* folder in the same localization;

  The file should contain number of rows and columns of the matrix (separated by spacebar) in its first line and then, an empty line. The next lines should contain further rows of the matrix. Each value in a row should be separated by spacebar.

- the expression given by the user should be correct in terms of matrix dimensions; the program does not detect mismatching matrix dimensions and if they are not correct, it will not work properly.

The program uses following functions:

- *read_matrix(file_name)* - which gets a file name without file extension (in general – relative path to a file) and returns a two-dimensional list which represents the matrix in a program;
- *print_matrix(matrix)* – which gets a two-dimensional list representing a matrix and prints it to the text file named *result.txt*;
- print_matrix_names(matrix_names) – which gets a list of names of the matrices (names of its source files) and prints them to the screen (it is used in a function that generates expression);
- *read_matrices(matrix_names)* – which gets an empty list of matrix names; it returns the list of matrices entered into the program and fills the list of names; the function uses the *read_matrix* one and acts as an interface for the user with importing matrices into the program;
- *print_math_operations()* – prints the available operations; it is used in a function that generates complete expression;

- *mx_multiplication(mx1, mx2)* – which gets two two-dimensional lists representing matrices and returns a two-dimensional list that is the result of their multiplication;
- *mx_adding(mx1, mx2)* – gets the same as *mx_multiplication* but returns the result of adding its parameters;
- *configure_expression(matrix_names, matrices)* – gets the list of matrix names (filled by *read_matrices* function*)* and list of matrices imported into the program (returned by *read_matrices* function); the function act like an interface for user with generating the desired expression that the program should solve; it returns a list which consist of two lists: multiplication and adding one. The lists are grouping the matrices in a way that lets the program solve the expression in correct sequence;
- *calculations(multiplication_list, adding_list)* – which gets the list returned by *configure_expression* function; that function is aimed to solve the expression using *mx_multilicating, mx_adding* functions and properly algorithm.

The program is intuitive to use. The user has to enter informations into the console and validate them by clicking 'Enter'. The program sends the user properly requests, informing him, what type of information it needs.

2. Virtual Machine

The cloud computer that was used for the project is an EC2 instance available on Amazon AWS Cloud.

The instance type is t3.medium. Main informations about the instance:

- full name: Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type
- vCPUs: 2
- Memory (GiB): 4
- Instance Storage: EBS.

For transferring necessary files between local and virtual computer, the *WinSCP* program was used. The machine is controlled with the use of *PuTTY* program. The aforementioned programs was used because of the Windows OS running on the local computer.

The connection between local and virtual computer is established using the private key that is always compared to the public key stored in a cloud (a key pair was generated at the end of creating a EC2 instance).

The program is launched in the cloud by typing *python* command and next, the path to .py file with code. The command can be used because of previously installed python environment. For that case, command: *sudo yum install python* was used.

3. Testing the program

The program was tested on a virtual machine by importing into it six matrices (named: m1, m2, m3, m4, m5, m6) and setting up an expression:

$$m1 \cdot m2 + m3 + m4 \cdot m5 \cdot m6$$

The matrices have following dimensions (row, col): m1(4, 8), m2(8, 4), m3(4, 4), m4(4, 8), m5(8, 4) and m6(4, 4). Their .txt files are included to this report. The result of that expression was compared using MATLAB.

4. Main task

The main task of that project is to run that program a few times on a EC2 instance and measure the time of calculations. Next, the results will be presented with a table and a chart.

Time measuring has been done with the use of *time.clock()* function of python language. At the beginning and end of *calculations* function, the variables *t1* and *t2* are assigned to the value returned by *time.clock()* function. The time that the EC2 has spent on calculations is evaluated as a difference between *t2* and *t1*. Next, the result is printed on the screen.

For the purpose of main task, there was used the expression form Section 3. The program was launched 5 times. For each time, there is only one matrix processed during calculations. It means that, the expression should now looks like:

$$m1 \cdot m1 + m1 + m1 \cdot m1 \cdot m1$$

The matrices contain random integer values in range form 0 to 10.

The dimensions of matrix *m1* for each test are:

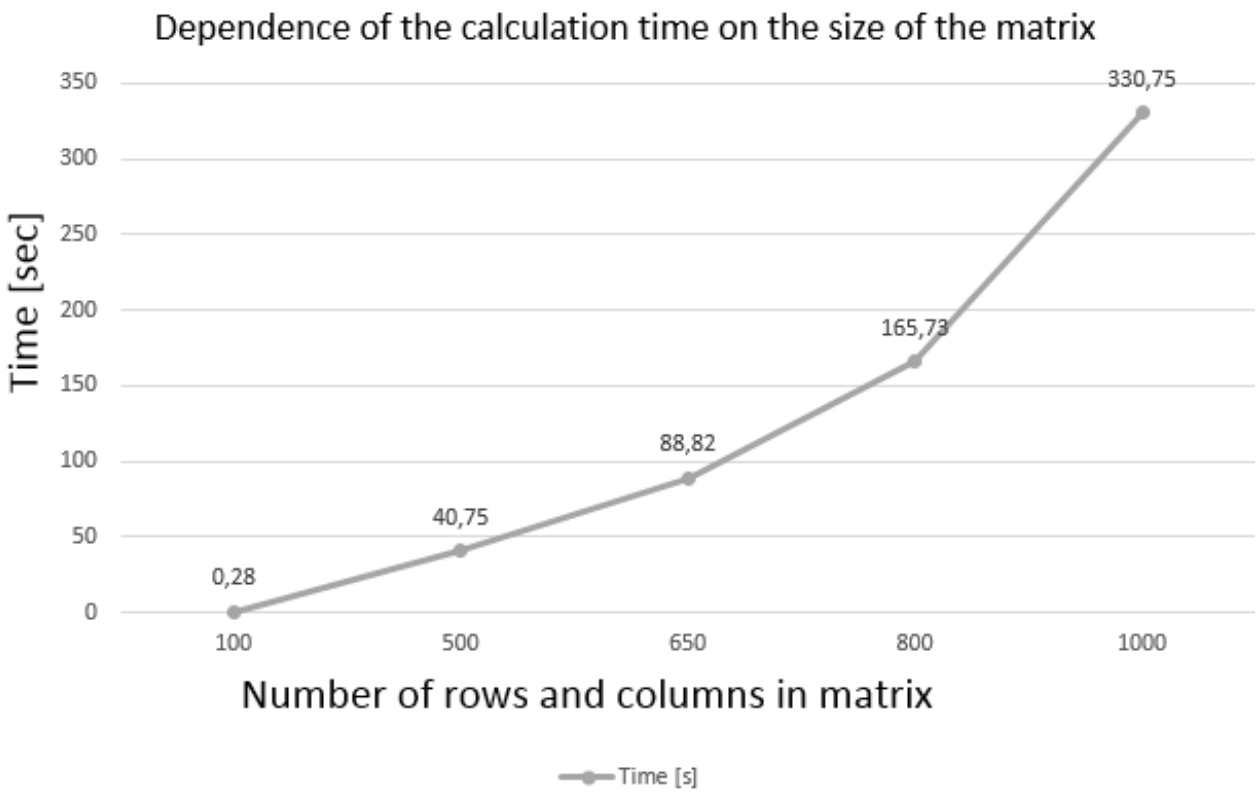
- first test – 100x100
- second test – 500x500
- third test – 650x650
- fourth test – 800x800
- fifth test – 1000x1000

5. Results of the project

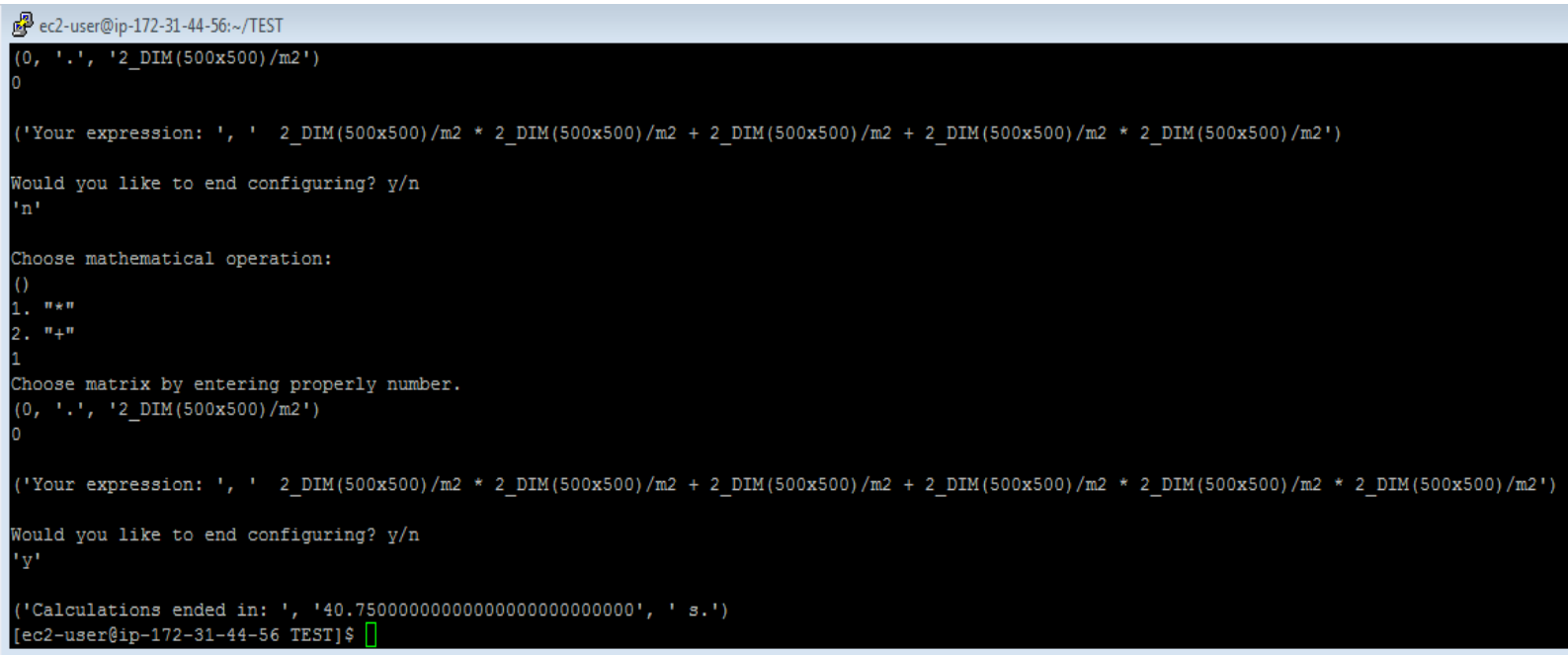
The table below describes the time in seconds that the used instance of EC2 needed to solve the expression each time.

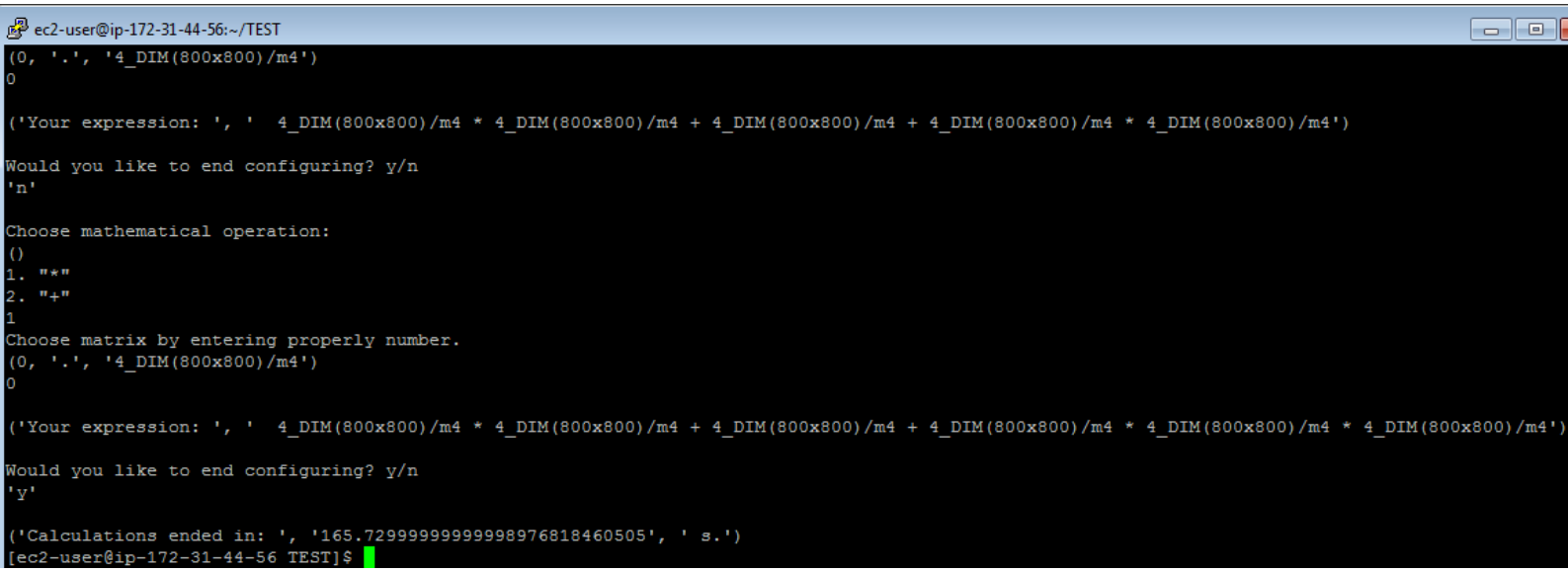
Times of calculations (s)	
m1(100x100)	0,28
m2(500x500)	40,75
m3(650x650)	88,82
m4(800x800)	165,73
m5(1000x1000)	330,75

The same data is presented on the chart below.



The screens below show examples of time results showed in EC2 console.





6. Additional Comments

While entering informations in EC2 console, it is important to remember that every string should be placed between ‘ ‘ characters.

Whole files are attached to this report. There is an *CloudProject* folder which includes *matrix_calculations.py* file with the code and *matrices* folder with .txt files of matrices. In that folder, there are six subfolders*: TEST, 1_DIM(100x100), 2_DIM(500x500)* … which contains of .txt files with matrices that were used for each test during that project.