Documentation-App Repairs

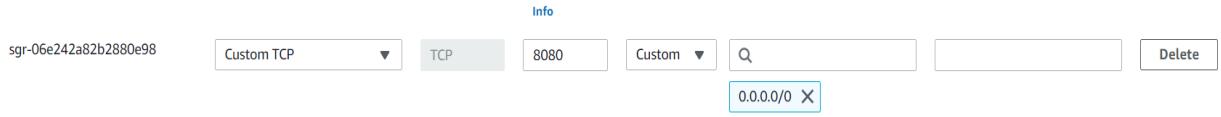
I utilized React and Django, two widely used web development frameworks, however they have diverse uses and are applied to various areas of a web application to create this application.

I will provide you the instructions for starting my application:

1. Getting your work environment ready

1.1.There are 2 possibilities, to clone the repository from git locally , for that you need to download pycharm idea from:https://www.jetbrains.com/pycharm/ or to create an account on https://portal.aws.amazon.com/billing/signup#/start/email , this will take some time because the websites wants that you to confirm your identity so you will need to use your credit card informations, this process will take two-six days

1.2. After you login into you amazon account you need to create in EC2 a new instance that use unix. After you create your instance you need to edit the inbounds roules from your Security Group in such way that you add a



1.3. You need now to create a new environment, so you go to Cloud9 section and create a new environment, it will take a few minutes.

2.Getting the project from git and Run some terminal commands

You go in your pycharm and open a new folder where you want the project to be cloned or use Cloud9 Env.From now on the steps will be almost identical:


```
>git clone hhtp://github.com/MrGordon112/last_chance

>cd last_chance

>pip install -r requirements

>sudo yum update

>sudo yum upgrade

>pip install djangorestframework

>pip install gunicorn

>npm install -g heroku

>pip install dj-database-url

>npm install

>python manage.py migrate

>cd front-end

>npm run build

>cd ..

>python manage.py runserver
```

Tech stack choises made :

For database I used PostGreSQL :

-is a trusted option for essential applications due to its stability and reliability.

-with the help of PostgreSQL's strong SQL capabilities, I dealed with the enourmous data input .

Why Python and Django?

- Python’s syntax is easy to read and to undersand, making it friendly for learning new concepts and reducing development time.

-Django, the web framework that use Python as programming language , provides pre-build components and tools that accelelate web application development(like the integrated db(sqlite))

Why Git?

Git is a version control that helps you to keep track of changes made to your code over time.It allows you to view, compare, and revert to previous versions of your files easily.

I choosed Git because is the most popular.

Why Cloud9 AWS ?

Becaused I used it before for Ruby on Rails projects but unfortunately has a lot of depreciated tools, and I faced a lot of errors, also the resources that are given for free(cpu and ram memory) are not enough because it happened many times that my environment to freeze

Why deploy on Heroku?

It was one of the only websites I worked with before, but now they add automatically a postgres database that costs 5 dollars per month to function.

What is gunicorn?

Gunicorn is a Python server that helps run your web application.It sits between your web server (like Nginx or Apache) an your Python code.Its main job is to handle incoming web requests and send back the appropriate responses.

The stepts to initialize the nginx server

1.Create a aws account

2.Go to Ec2 instances and create a new Security Groupt with 3 new inbound rules

Ssh port 22 0.0.0.0/0

http port 80 0.0.0.0/0

http port 80 ::/0

3.Lounch instance and select Ubuntu and your Security Group that you just created.

4.Connect to the instance and run:

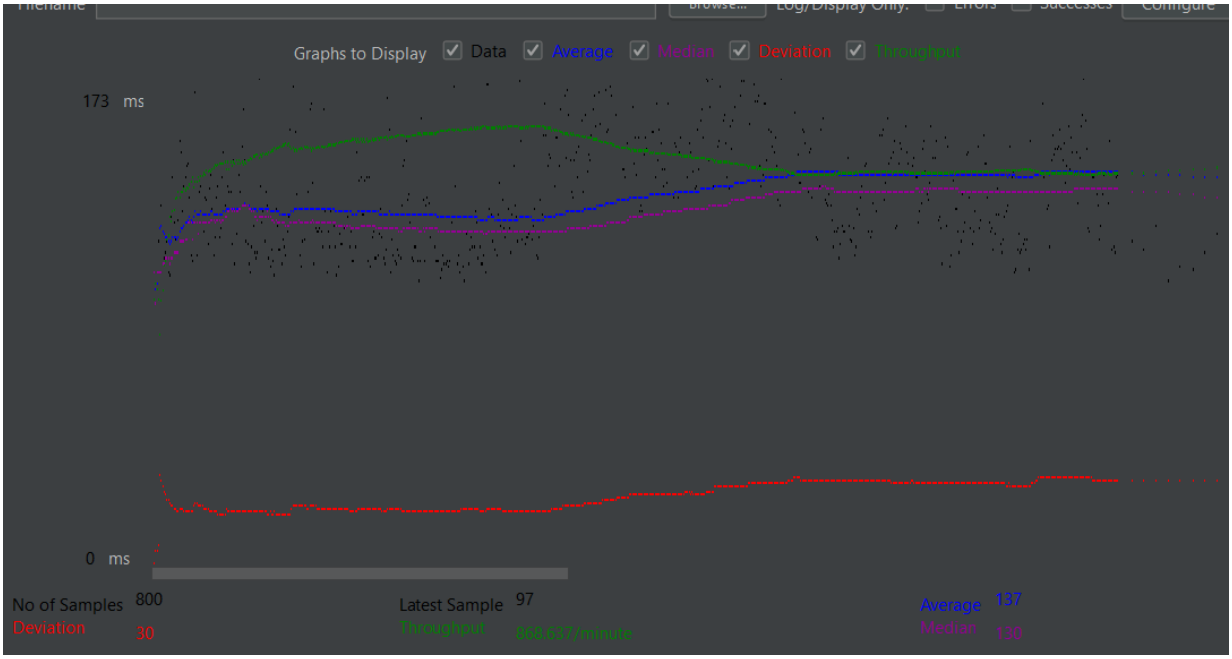
```
>clear

>sudo apt-get update

>sudo apt-get upgrade

Press enter when you see the restart alert
```

```
>sudo apt-get install python3-venv

>python3 –m venv env

>source env/bin/activate

>pip3 install Django

>git clone hhtp://github.com/MrGordon112/last_chance

>sudo apt-get install –y nginx

>pip install gunicorn

>sudo apt-get install supervisor

>sudo touch gunicorn.conf

>sudo nano gunicorn.conf

                [program:gunicorn]

                Directory = /home/Ubuntu/elevate

                Commnd= /home/Ubuntu/env/bin/gunicorn –workers 3 –bind unix:/home/Ubuntu/elevate/
app.sock my_django_app.wsgi:application

                autostart = true

                startsecs = 0

                autorestart = true

                stders_logfile=/var/log/gunicorn/gunicorn.err.log

                setout_logfile=/var/log/gunicorn/gunicorn.out.log

                [group:guni]

                programs:gunicorn
after this you need to save the file with ctrl+o , enter , ctrl + x



>sudo mkdir/var/log/gunicorn
```

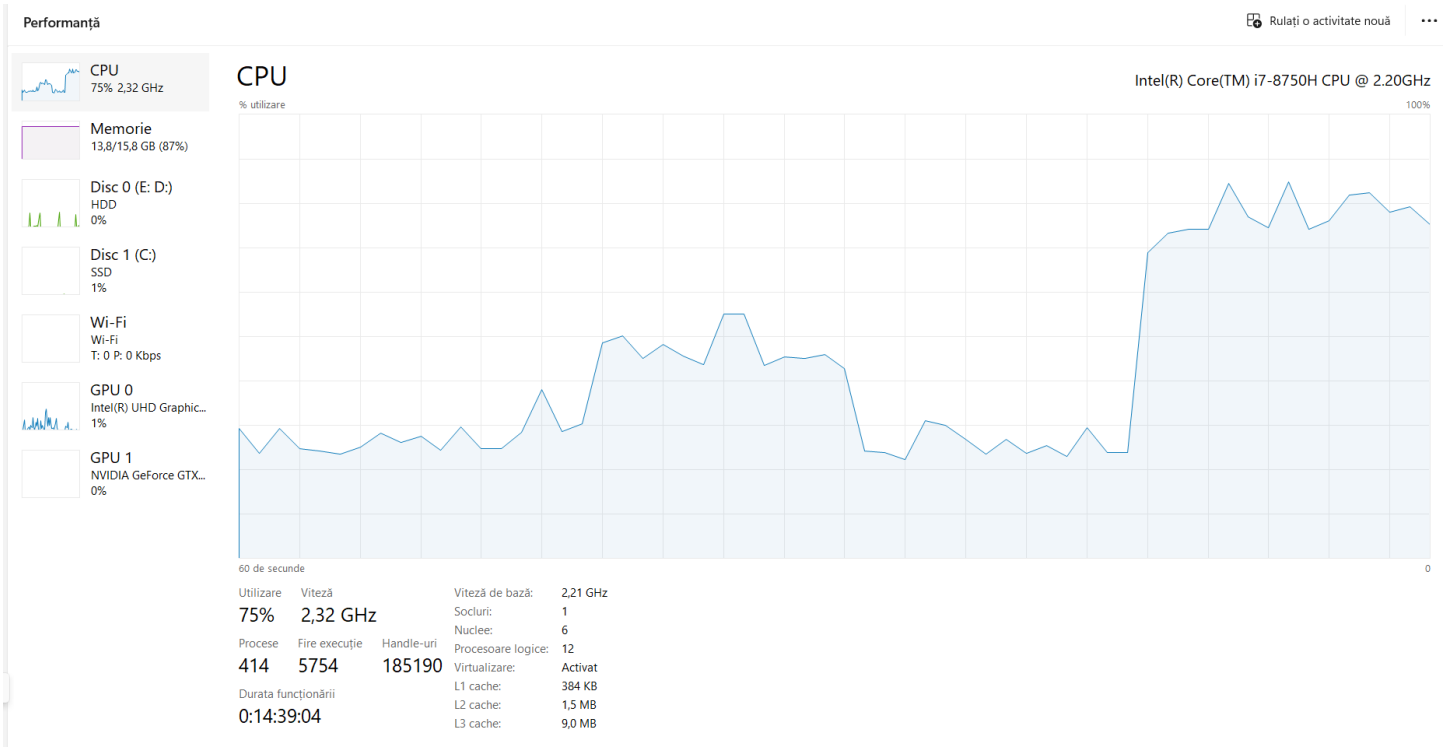
```
>sudo supervisorsorctl reread

>sudo supervisorctl update

>cd ..

>cd ..

>cd nginx

>sudo nano nginx.conf (here you change www-data to your root)

Save and exit (ctrl+o,enter,ctrl+x)

>cd sites-available

>sudo touch django.conf

>sudo nano django.conf

        server{

                listen 80

                server_name url

        location/{

                include proxy_params;

                proxy_params http://unix:/home/ubuntu/app.sock my_django_app

(save and exit)

>sudo nginx –t

>sudo ln django.conf /etc/nginx/sites-enabled

>sudo service nginx restart


That’s it, now you press on the ip that the bash will show you .
```

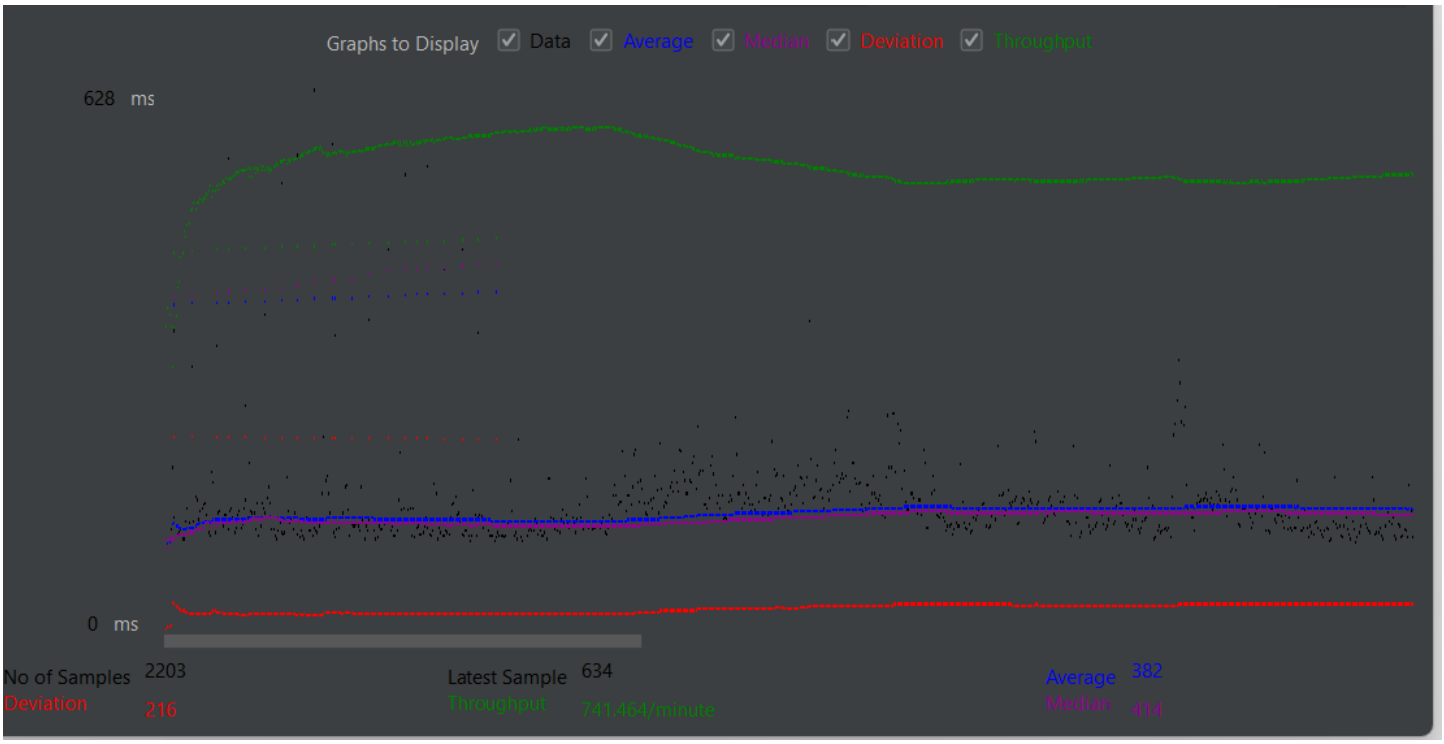
The spikes tests

With 2 users:

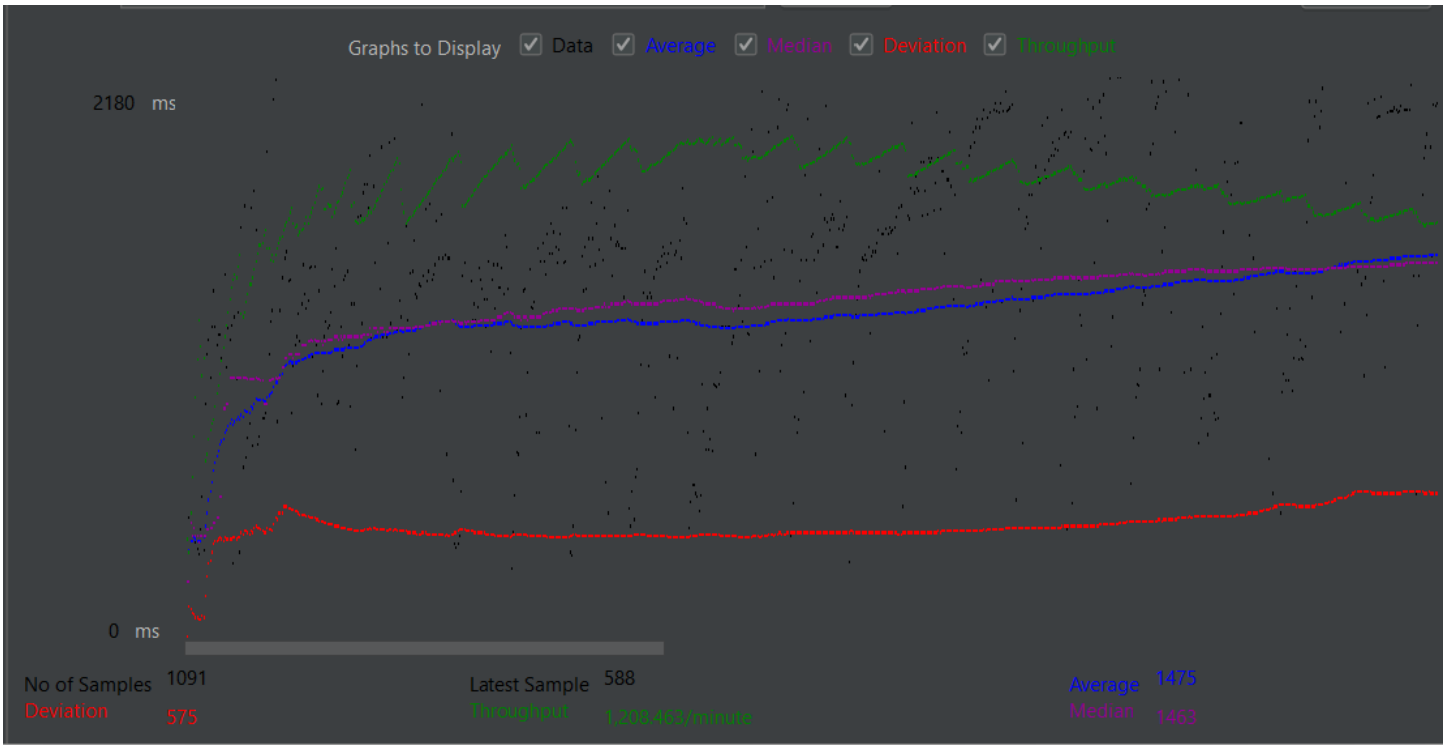


With 10 users

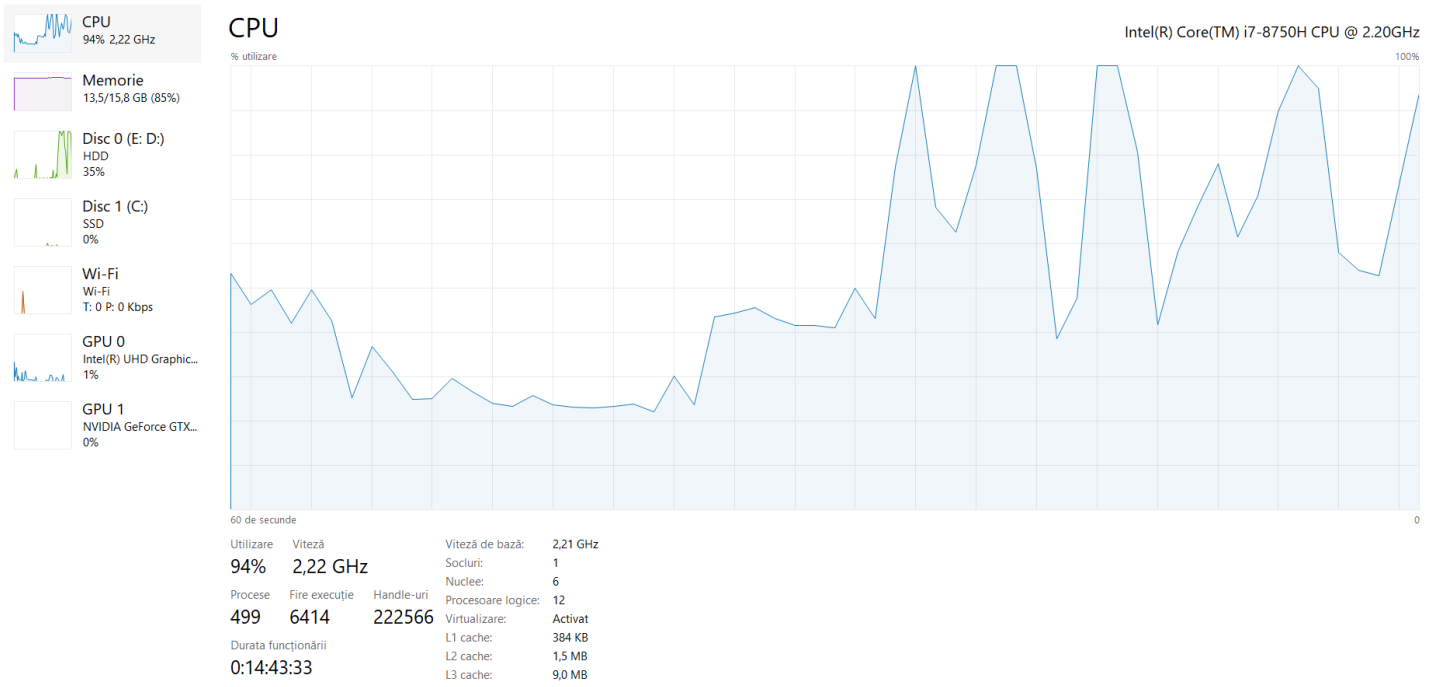


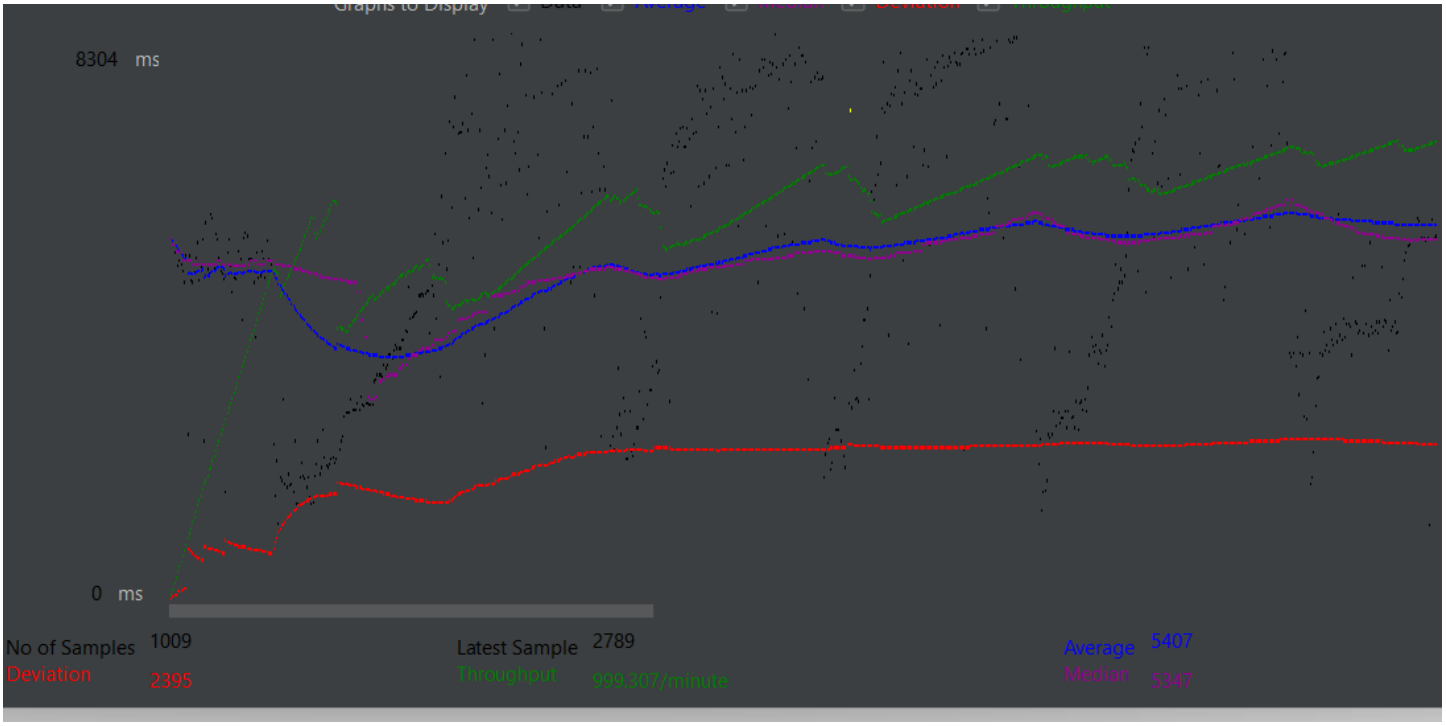


With 30 users

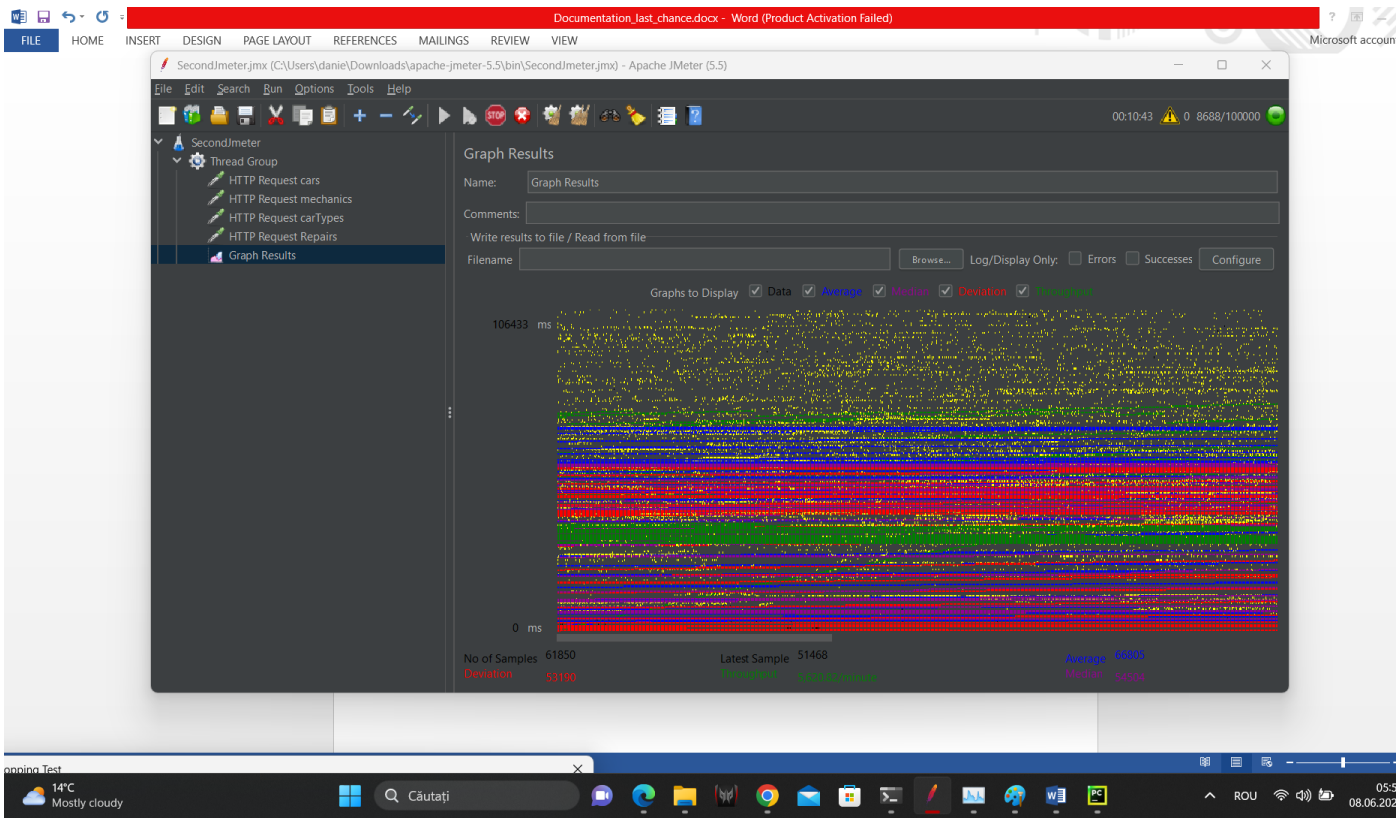


With 100 users





Cu 70.000



The Machine Learning implementation