

THESIS STATUS REPORT

WEEK 50 - 51

DONE

I started to implement the load testing sandbox. For that, I researched several Kubernetes and the Docker¹ SDK for python for the deployment of the microservice. I am still deciding whether I should use the official² or community³ Kubernetes SDK. For the GUI, I decided to use the package *PySimpleGUI*⁴ because it satisfies the simple needs for this GUI as it is mostly functional. I also did some research on the Extra-P⁵ python package and got to know how the input format works.

I decided to use the load testing framework called *locust*⁶ because it is based on python and usable as a library. This enables easier integration and automation. Furthermore, I can get the response time from it.

Now I am implementing the various methods that are needed to build a docker image from a given docker file to deploy it as a Kubernetes container. After the deployment is done, the load testing will take place, and finally, the resulting data will be converted such that the machine learning model and extra-p can handle it.

I uploaded the code temporarily to my private GitHub repository.

UPDATE FROM SYNC MEETING (18.12.2020)

We discussed that I should also try to implement a neural network and compare it to the Support Vector Regression approach. Furthermore I will get access to the university repository to upload my code in the future.

NEXT STEPS

1. Finish implementing the synthetic load testing sandbox (including research to all its components)

¹ <https://docker-py.readthedocs.io/>

² <https://github.com/kubernetes-client/python>

³ <https://github.com/mnubo/kubernetes-py>

⁴ <https://pysimplegui.readthedocs.io/>

⁵ <https://github.com/extra-p/extrap>

⁶ <https://locust.io/>