‘DeepRL4MEC’ experiment Review @ FEC9

Dr Iraklis Nikolakakos

Lamda Networks (ex. Modio)

FEC9

25 May 2021

LAMDA NETWORKS

Deep Reinforcement Learning for MPEG DASH segment dynamic assignment in MEC environments

DEEPRL4MEC

Experiment description

BACKGROUND AND MOTIVATION

Mobile Edge Computing (MEC) can significantly improve the QoS of services; however, the computing resources at MEC nodes are limited and should be efficiently allocated.

Within our stage-1 experiment, we tackle the problem of how to best allocate resources within one MEC node so that the offered video services achieve the best possible performance.

Experiment description

CONCEPT

Our approach - DeepRL4MEC - learns the pattern of jobs entering the MEC system.

By learning the pattern of jobs, DeepRL4MEC **avoids both the phenomena of the pre-emption of small jobs and the pre-emption of large jobs**.

Its ability to learn avoids the drawbacks of e.g. random and the SJF methods.

Experiment description

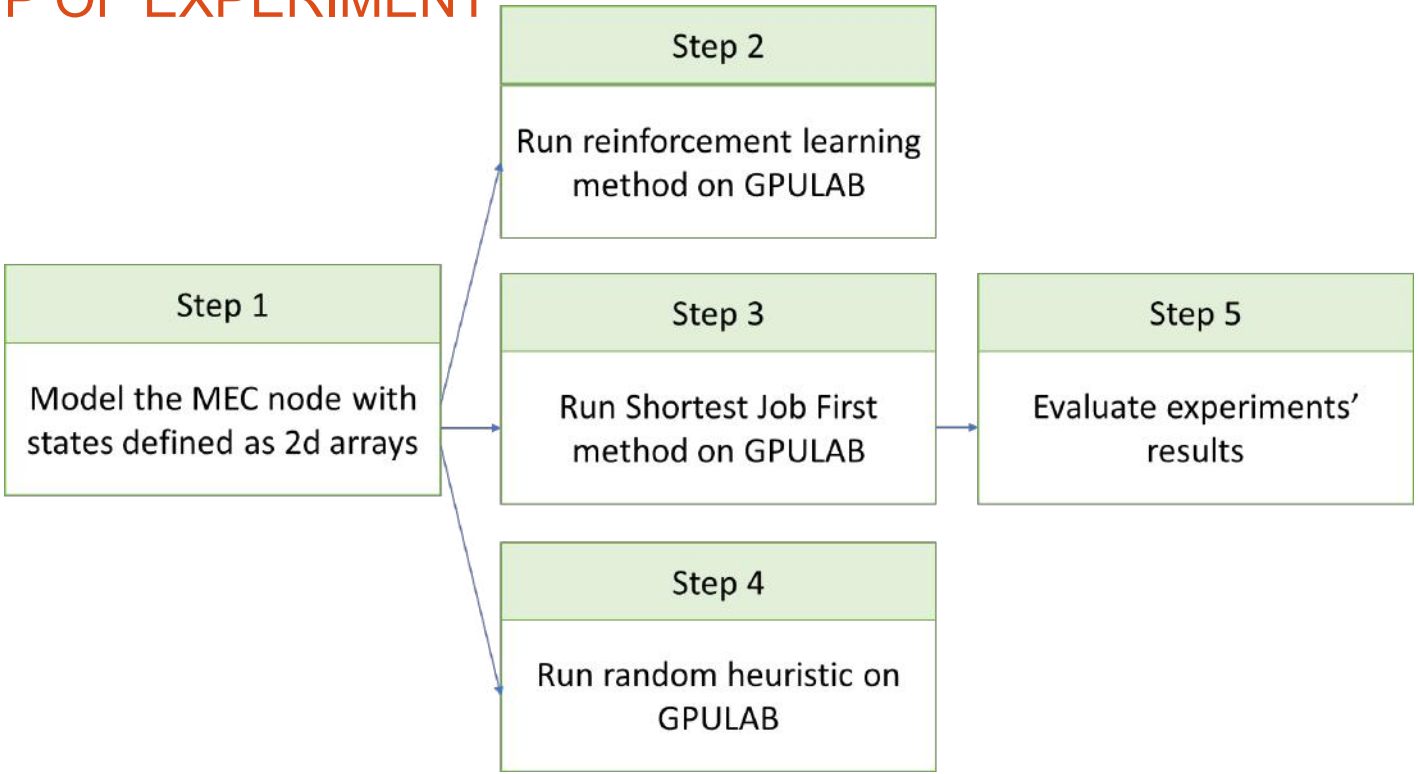
OBJECTIVES

Objective of Stage 1

Rewrite of parts of our decision-making algorithm in order to be able to undertake decisions per MPEG DASH segment, not for a whole video file requested by a MPEG DASH client and test of the new code using GPUs drawn from the GPULAB testbed.

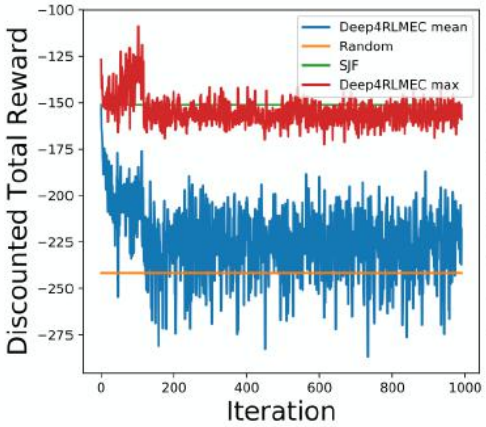
Experiment description

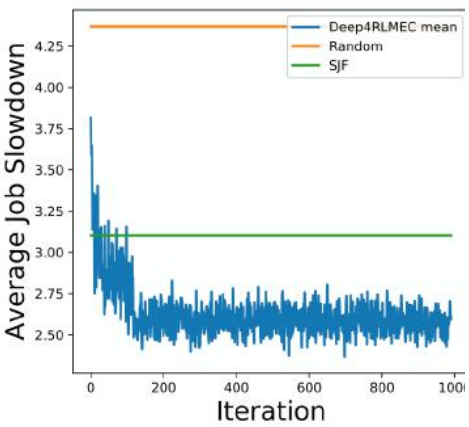
SETUP OF EXPERIMENT

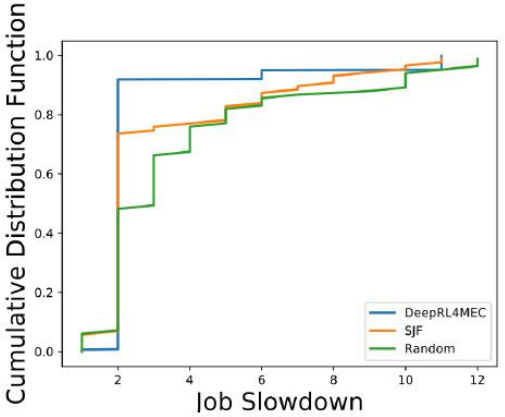


Project results

MEASUREMENTS {PROB SMALL JOB=0.1, LAMDA OF JOB ARRIVAL =0.9}

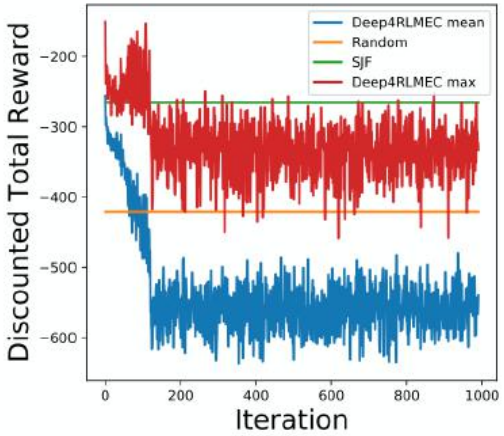


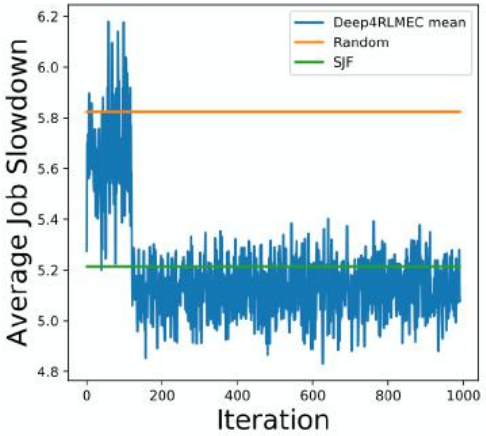


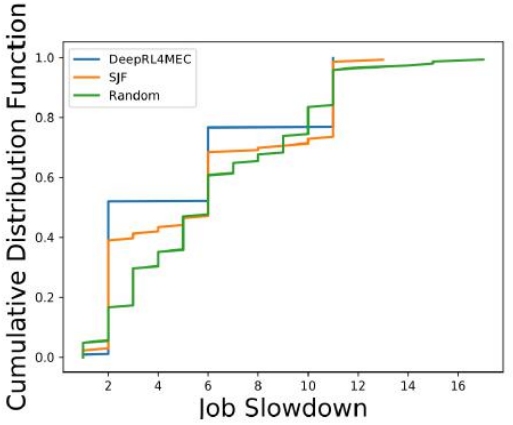


Project results

MEASUREMENTS {PROB SMALL JOB=0.5, LAMDA OF JOB ARRIVAL =0.9}







Project results

LESSONS LEARNT

Our experimentation within Fed4FIRE+ has provided us with significant knowledge on how to adapt our code by parallelizing certain functions in order to be able to assign tasks to arrays of GPUs within IMEC's GPULAB.

Business impact

IMPACT ON BUSINESS

The knowledge gained from FED4FIRE+ empowers us to pursue our strategic roadmap in creating a deep reinforcement learning provisioning engine for MEC environments.

Business impact

HOW FED4FIRE+ HELPED US

The most important benefit was that we used resources – otherwise very expensive to use - from GPULAB for running our machine learning algorithm and Virtual Wall 2 in order to obtain access to the results produced by our GPULAB experiments.

We proved that use of a GPU environment can significantly shorten the training time for large sets of training data: by parallelizing our code and employing arrays of GPUs, we have seen that training CNN networks can be realised in the order of hours and not in the order of days, which was the case when we were using our code on plain CPUs in our lab.

Business impact

VALUE PERCEIVED

A GPU array approach with parallel code handing different training experiments and then combining their results can lead to a viable system that can be implemented in an real MEC setting.

Business impact

WHY WE CAME TO FED4FIRE+

Our Fed4FIRE+ stage-1 experiment was designed following our strategic decision to grasp the 5G/6G market opportunity and innovate within the mobile edge market segment.

We applied to FED4FIRE+ for its GPULAB offering and our prior excellent experience with FED4FIRE+.

Feedback

USED RESOURCES

[Virtual Wall] Two physical nodes. One in Virtual Wall 1 in order to run some CPU-base experiments and one in Virtual Wall 2, in order to sftp the results obtained from GPULAB in the project's shared folder.

[GPULAB] We used various nodes within GPULAB during different experiments.

Feedback

TOOLS

[jFed] We used this tool to obtain results from GPULAB from the shared project space.

[gpulab-cli] We used this tool to submit jobs, SSH to the docker container, install extra libraries when logged in to the container and for seeing the debugging messages of our code as well as the GPU usage with the command ‘nvidia-smi’.

[https://gpulab.ilabt.imec.be] We used this web tool to cancel jobs that were no longer needed during our experiments. We also used the tool to see the usage in terms of memory, CPU and GPU resources.

Feedback

SATISFACTION FROM FED4FIRE+ TESTBED OFFERING

The GPULAB resources we employed for the purposes of our work were sufficient and we also were very satisfied from the CPU/GPU characteristics and performance of the nodes we have been reserving within GPULAB.

Feedback

DOCUMENTATION AND SUPPORT

The documentation from https://doc.ilabt.imec.be/ilabt/gpulab/index.html was very clear and complete. We did not experience any issues since the documentation includes very clear instructions about how to run an experiment over GPULAB.

We are overall very satisfied with the support from helpdesk@ilabt.imec.be and also with our patron Pieter Becue. The responses to our emails were very fast.

Experiment description

OBJECTIVES

Objective of Stage 2

- Use the stage-1 software base and expand it to cater for the distributed flavor of the problem, i.e. how to use reinforcement learning theory and deep neural networks methods in order to minimize user-defined objective functions within multi-PoP MEC environments.
- Finally, to validate the expanded code base on GPULAB.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, which is co-funded by the European Commission and the Swiss State Secretariat for Education, Research and Innovation, under grant agreement No 732638.

WWW.FED4FIRE.EU