

# Configuration Manual - Hybrid Reinforcement Learning based code offloading in MEC

MSc Research Project  
Cloud Computing

Abhinash Pati  
Student ID: 18189440

School of Computing  
National College of Ireland

Supervisor: Manuel Tova-Izquierdo

National College of Ireland  
Project Submission Sheet  
School of Computing



<b>Student Name:</b>	Abhinash Pati
<b>Student ID:</b>	18189440
<b>Programme:</b>	Cloud Computing
<b>Year:</b>	2018
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Manuel Tova-Izquierdo
<b>Submission Due Date:</b>	20/12/2018
<b>Project Title:</b>	Configuration Manual - Hybrid Reinforcement Learning based code offloading in MEC
<b>Word Count:</b>	XXX
<b>Page Count:</b>	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

<b>Signature:</b>	
<b>Date:</b>	17th August 2020

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project</b> , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual - Hybrid Reinforcement Learning based code offloading in MEC

Abhinash Pati  
18189440

## 1 Introduction

The scope of the manual is limited to provide a guideline to the proposed MSc Research project in order to replicate the presented results. The project has been developed in python 3.7. All the required modules, VM configurations as well installation procedure have been listed in this document.

## 2 Pre-requisites

The project have been developed using python 3.7 and python setup is mandatory to run the project.

The project is deployed in a virtual machine in Microsoft Azure and the results shown are obtained from running the python script on that machine, however, the results with some minor variation can also be generated with any local machine having python installed.

### 2.1 Creating a virtual machine in Azure

An Azure virtual machine can be created directly via the Azure portal. The portal can be accessed through any browser by login into <https://portal.azure.com>

The following are the configurations and steps required to create a Linux based VM machine in Azure.

#### 2.1.1 Configuration

Table 1: VM Configuration Parameters

Parameters	Assigned value
OS	Ubuntu 18.04-LTS
VM Type	Standard A2m_v2
vCPUs	2
Ram	16 Gb
Disk	30 Gb
Disk Type	Premium SSD
Region	East US

### 2.1.2 VM Setup

1. In the search bar, search virtual machines and select virtual machines.
2. In the virtual machines page, click Add to open create a virtual machine page.
3. Click on Create new resource group. Type anyNameForTheGroup.
4. In the region, select East US and choose Ubuntu 18.04 LTS as OS Image. Choose Standard A2m.v2 with 16gb ram and 2 vCPUs as VM size and leave the other defaults as shown in Fig 1.

Home > Virtual machines >

## Create a virtual machine

Subscription \* ⓘ Azure for Students

Resource group \* ⓘ (New) anyNameForTheGroup\_group  
[Create new](#)

### Instance details

Virtual machine name \* ⓘ anyNameForTheGroup ✓

Region \* ⓘ (US) East US

Availability options ⓘ No infrastructure redundancy required

Image \* ⓘ Ubuntu Server 18.04 LTS - Gen1  
[Browse all public and private images](#)

Azure Spot instance ⓘ  Yes  No

Size \* ⓘ Standard\_A2m\_v2 - 2 vcpus, 16 GiB memory (€73.26/month)  
[Select size](#)

[Review + create](#) [< Previous](#) [Next : Disks >](#)

Figure 1: Region-OS-VM Type Configuration

5. Follow the steps in Watson et al. (2020) to create an SSH connection to connect to the above VM.

### 2.1.3 Python Setup

The VM configured with Ubuntu 18.04 or above usually comes with python pre-installed. We need to install another python environment separately so that the modules which are dependent on the old version of python is not hindered Simic (2020). Use **python** —**version** to check the installed version.

1. Open a terminal and update apt  
**sudo apt update**

2. Install libraries required for python  
**sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget git**
3. Download python 3.7's source code  
**mkdir tmp**  
**cd /tmp**  
**wget https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz**
4. Extract the downloaded source code zip file  
**tar -xf Python-3.7.5.tgz**
5. Enable python optimization to run python scripts faster.  
**cd python-3.7.5**  
**./configure --enable-optimizations**
6. Install a separate instance of python  
**sudo make altinstall**
7. Verify installed version  
**python3 --version**

## 3 Running the research project

To run the research project, make sure all the pre-requisites mentioned in section 2 are installed and configured.

Before running the simulation of the research project, we need to install the python modules used in the project.

### 3.1 Python modules

Save the following list of modules into a text file called pip install requirements.txt

```
absl-py==0.9.0
astunparse==1.6.3
attrs==19.3.0
backcall==0.2.0
bleach==3.1.5
cachetools==4.1.1
certifi==2020.6.20
chardet==3.0.4
cyclerc==0.10.0
decorator==4.4.2
defusedxml==0.6.0
entrypoints==0.3
gast==0.3.3
geographiclib==1.50
geopy==2.0.0
google-auth==1.19.2
google-auth-oauthlib==0.4.1
```

google-pasta==0.2.0  
grpcio==1.30.0  
h5py==2.10.0  
idna==2.10  
ImageHash==4.1.0  
importlib-metadata==1.7.0  
ipykernel==5.3.3  
ipython==7.16.1  
ipython-genutils==0.2.0  
ipywidgets==7.5.1  
jedi==0.17.2  
Jinja2==2.11.2  
jsonschema==3.2.0  
jupyter==1.0.0  
jupyter-client==6.1.6  
jupyter-console==6.1.0  
jupyter-core==4.6.3  
Keras-Preprocessing==1.1.2  
kiwisolver==1.2.0  
Markdown==3.2.2  
MarkupSafe==1.1.1  
matplotlib==3.3.0  
mistune==0.8.4  
nbconvert==5.6.1  
nbformat==5.0.7  
notebook==6.0.3  
numpy==1.19.0  
oauthlib==3.1.0  
opencv-python==4.3.0.36  
opt-einsum==3.3.0  
packaging==20.4  
pandas==1.0.4  
pandocfilters==1.4.2  
parso==0.7.0  
pexpect==4.8.0  
pickleshare==0.7.5  
Pillow==7.2.0  
prometheus-client==0.8.0  
prompt-toolkit==3.0.5  
protobuf==3.12.2  
ptyprocess==0.6.0  
pyasn1==0.4.8  
pyasn1-modules==0.2.8  
Pygments==2.6.1  
pyparsing==2.4.7  
pyrsistent==0.16.0  
python-constraint==1.4.0  
python-dateutil==2.8.1

```
pytz==2020.1
PyWavelets==1.1.1
pyzmq==19.0.1
qtconsole==4.7.5
QtPy==1.9.0
requests==2.24.0
requests-oauthlib==1.3.0
rsa==4.6
scipy==1.4.1
Send2Trash==1.5.0
six==1.15.0
tensorboard==2.2.2
tensorboard-plugin-wit==1.7.0
tensorflow==2.2.0
tensorflow-estimator==2.2.0
termcolor==1.1.0
terminado==0.8.3
testpath==0.4.4
tornado==6.0.4
traitlets==4.3.3
urllib3==1.25.9
wcmwidth==0.2.5
webencodings==0.5.1
Werkzeug==1.0.1
widgetsnbextension==3.5.1
wrapt==1.12.1
zipp==3.1.0
```

Once that's done, install all the modules into the python environment.

```
pip install -r /path/to/requirements.txt
```

## 3.2 Code setup

Extract the zip file of the code with in the home directory with the main project folder name as research. The path for dataset needs to be set in files: Parameters.py, DQN.py and optimization.py

## 3.3 Starting the simulation

One can run the simulation by using the following command:

```
python3 simulation.py
```

However, the default parameters defined in parameter.py can take upto 48hrs to finish the simulation. A shorter simulation can be created by changing the values of Number of Time Slots and Time Slot Length in parameter.py.

One can also run the simulation as background and generate the results after the background process is finished. To do so:

```
chmod +x simulation.py
```

```
nohup python3 simulation.py &
```

We can check if the simulation process started has ended or not via

```
ps ax — grep simulation.py
```

The output of the background process of simulation.py is stored in a text file **nohup.out** and can be seen by using

```
cat nohup.out
```

## 4 Graph generation

Once the simulation.py script have been completed, a text file called **overall\_costs.txt** will be generated that would consists all the cost of execution for my algorithm as well as the benchmarking algorithm. One can generate the graph by running:

```
python3 utils/diagram.py
```

If the scripts throws a file not found error, check the file path mentioned in diagram.py and rectify it accordingly to point to the text file containing the results i.e., overall\_costs.txt. If everything is followed as per the guide, a graph will be generated similar to Fig 2.

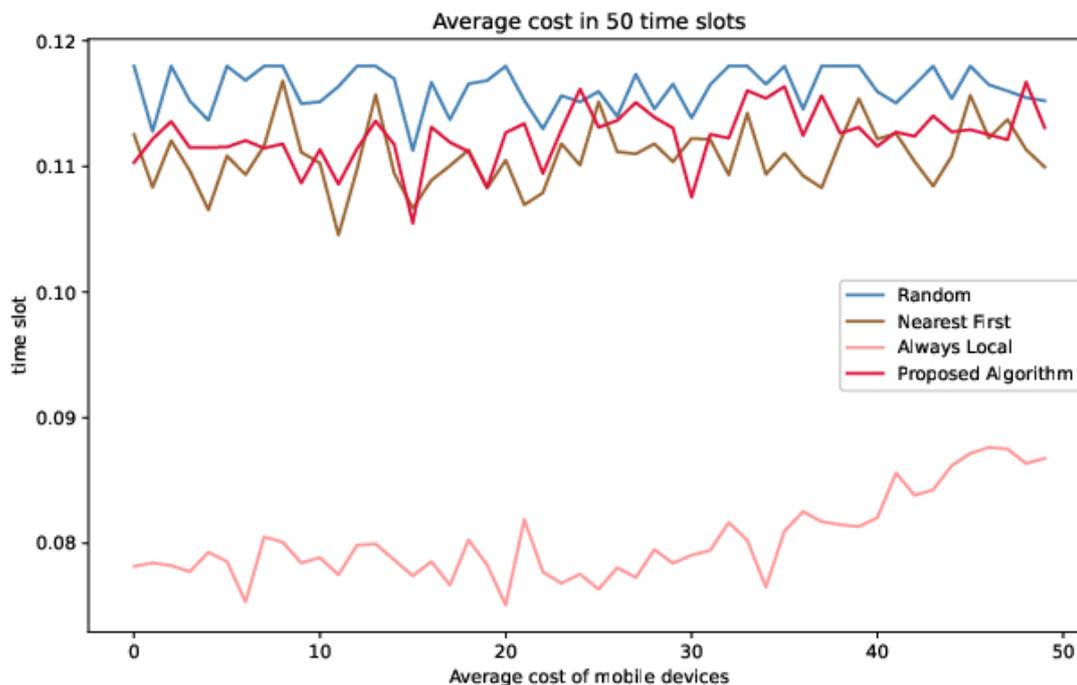


Figure 2: Region-OS-VM Type Configuration

## References

Simic, S. (2020). How to install python 3 on ubuntu 18.04 or 20.04.

URL: <https://phoenixnap.com/kb/how-to-install-python-3-ubuntu>

Watson, C., Foulds, I. and Berry, D. C. (2020). Quickstart: Create a linux virtual machine in the azure portal, *Technical Report 1*, Microsoft, US.