National College of Ireland

Configuration Manual

MSc Research Project
MSc Cloud Computing

Rajkumar Nagulsamy
Student ID: x22102817

School of Computing
National College of Ireland

Supervisor: Punit Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rajkumar Nagulsamy
Student ID:	x22102817
Programme:	MSc Cloud Computing **Year:** 2024
Module:	Research Project
Lecturer:	Punit Gupta
Submission Due Date:	16/09/2024
Project Title:	EP-MPCHS: Edge Server based cloudlet offloading using Multi-Core and Parallel Heap Structures
Word Count:	1310 **Page Count:** 8

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.

Signature: Rajkumar Nagulsamy

Date: 16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Rajkumar Nagulsamy
X22102817

1 Introduction

In the continuously developing world of Cloud Computing and mobile technologies computer demanding resource needs have witnessed a strong climb for mobile applications. The task – resource management has become a major research paradigm for mobile resource offloading. According to Roman et al. (2018), during the previous 10 years, cloud computing has evolved as one of the most compelling solutions for hosting applications via the Internet infrastructure. computer in the cloud is based on the usage of centralized computer resources hosted in data centers. This method alleviates the load of infrastructure management and allows service providers to deliver cloud services at charges that are fair for both consumers and enterprises.

The implementation is shown below

2 Prerequisites

Install python 3.11.4 and install

```
!pip install pandas
!pip install  numpy
!pip install  matplotlib


!pip install numpy==1.23.5
!pip install pandas
!pip install matplotlib
!pip install networkx
!pip install msgpack
```

3 Prerequisite Installation

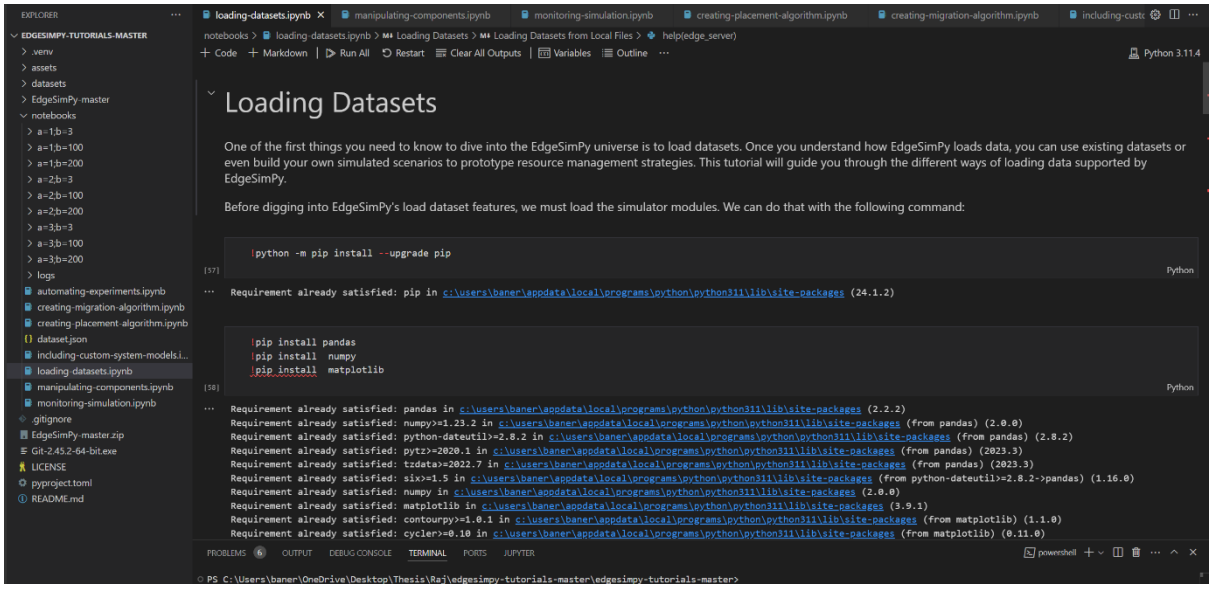
3.1 Installation of libraries

```
Collecting numpy==1.23.5
  Using cached numpy-1.23.5-cp311-cp311-win_amd64.whl.metadata (2.3 kB)
Using cached numpy-1.23.5-cp311-cp311-win_amd64.whl (14.6 MB)
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 2.0.0
    Uninstalling numpy-2.0.0:
      Successfully uninstalled numpy-2.0.0
Successfully installed numpy-1.23.5
Requirement already satisfied: pandas in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (2.2.2)
Requirement already satisfied: numpy>=1.23.2 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from pandas) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: tzdata>=2022.7 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: matplotlib in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (3.9.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.1.0)
Requirement already satisfied: cycler>=0.10 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (4.40.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.23 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=8 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: networkx in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (3.3)
Requirement already satisfied: msgpack in c:\users\baner\appdata\local\programs\python\python311\lib\site-packages (1.0.8)
```

3.2 Step 1: VSCODE installation

https://code.visualstudio.com/download

Step 2: Open Folder with VSCODE



4 Running the proposed Code

We need to run the codes in the following order to achieve the specific order. The Running of the proposed component.

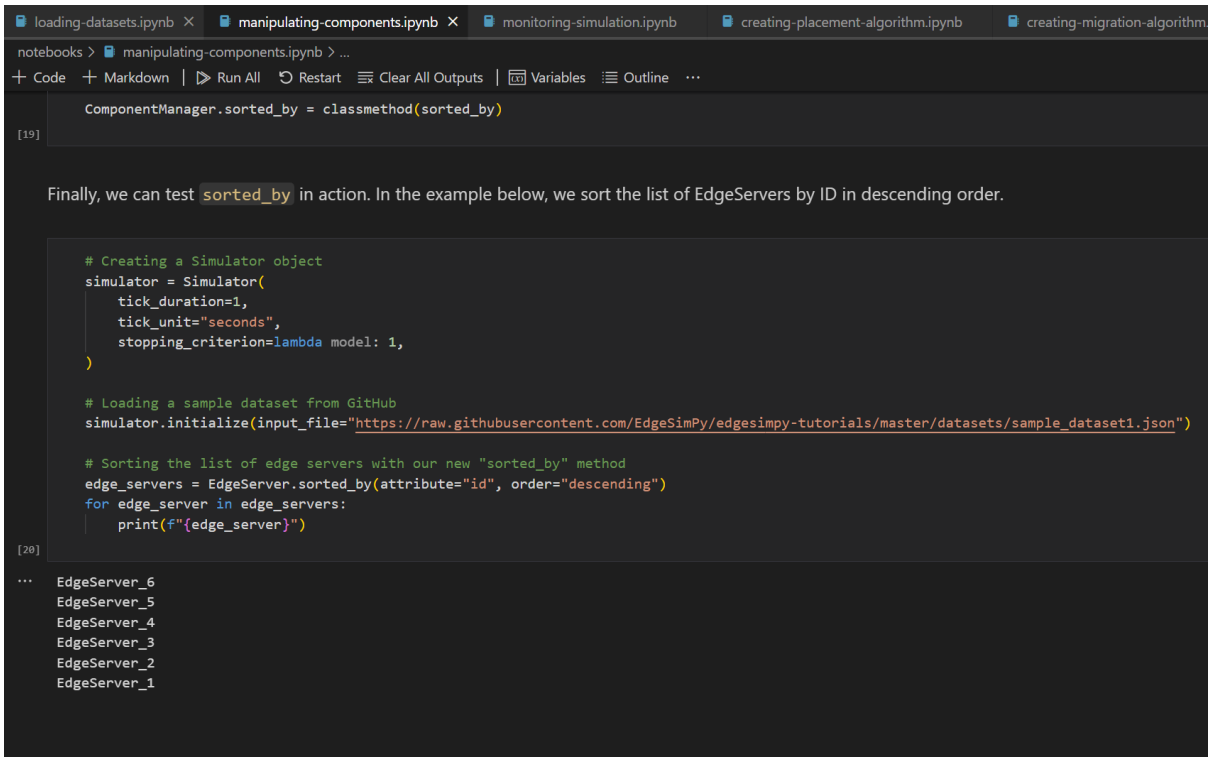
Tutorial
Loading datasets
Manipulating components
Monitoring simulation
Creating a placement algorithm
Creating a migration algorithm
Automating experiments
Using built-in and custom system models

4.1 Loading Dataset

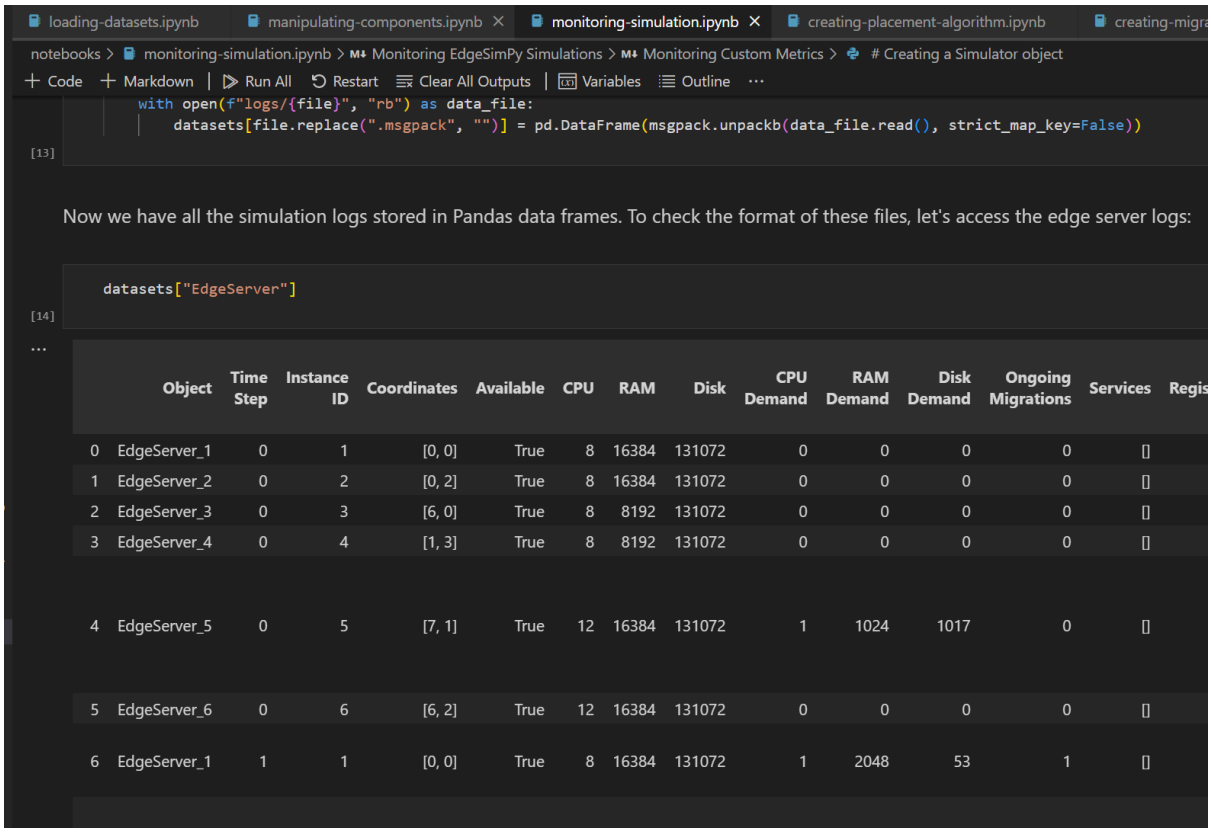
```
!curl https://raw.githubusercontent.com/EdgeSimPy/edgesimpy-tutorials/master/datasets/sample_dataset1.json --output dataset.json

# Creating a Simulator object
simulator = Simulator()

# Loading the dataset from the local "dataset.json" file
simulator.initialize(input_file="dataset.json")

# Displaying some of the objects loaded from the dataset
for edge_server in EdgeServer.all():
    print(f"{edge_server}. CPU Capacity: {edge_server.cpu} cores")
```

```
EdgeServer_1. CPU Capacity: 8 cores
EdgeServer_2. CPU Capacity: 8 cores
EdgeServer_3. CPU Capacity: 8 cores
EdgeServer_4. CPU Capacity: 8 cores
EdgeServer_5. CPU Capacity: 12 cores
EdgeServer_6. CPU Capacity: 12 cores
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
 55  142k   55 81920    0     0   707k      0 --:--:-- --:--:-- --:--:--  714k
100  142k  100  142k    0     0  1212k      0 --:--:-- --:--:-- --:--:-- 1222k
```

4.2 Manipulating components

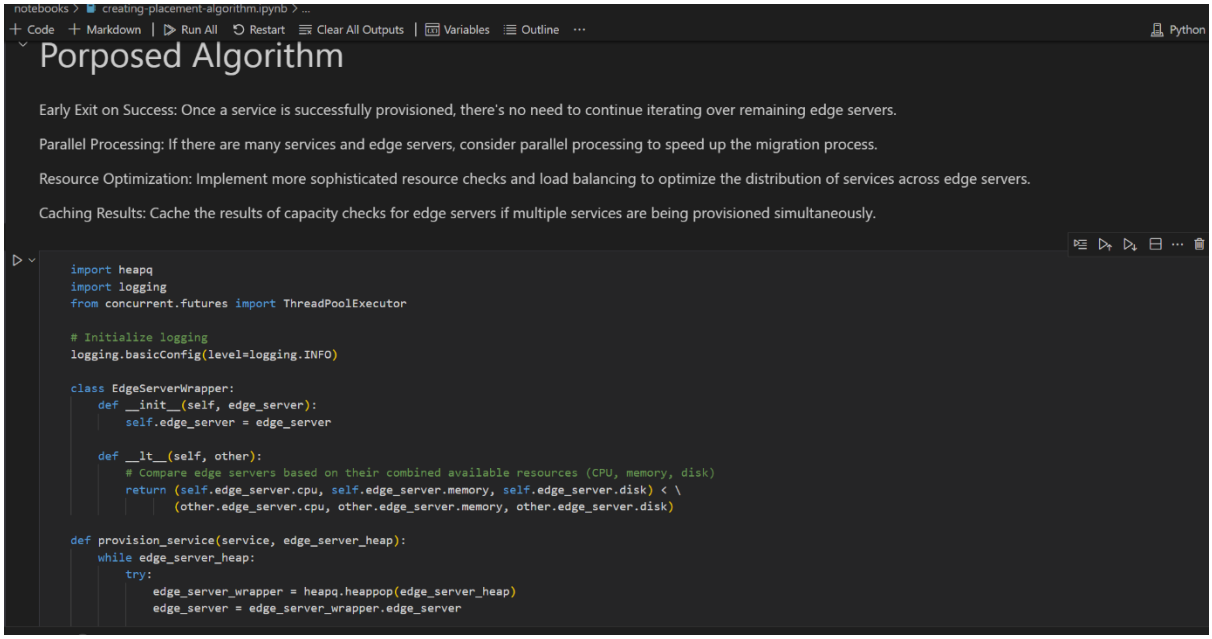


4.3 Monitoring Simulation

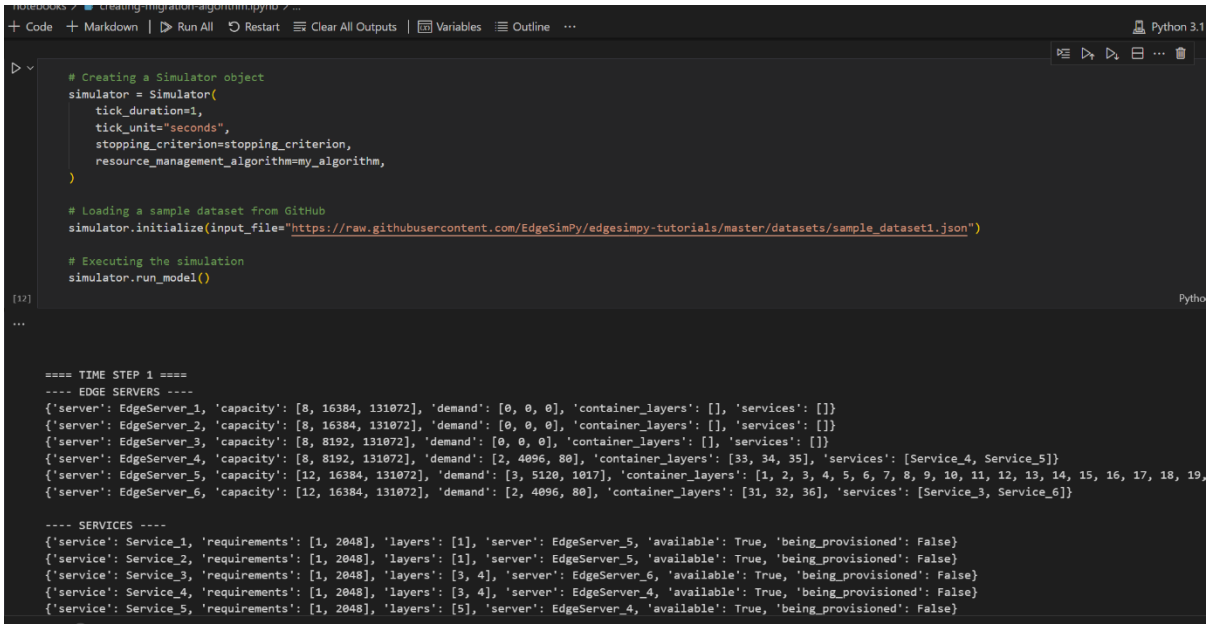


	Object	Time Step	Instance ID	Coordinates	Available	CPU	RAM	Disk	CPU Demand	RAM Demand	Disk Demand	Ongoing Migrations	Services	Regis
0	EdgeServer_1	0	1	[0, 0]	True	8	16384	131072	0	0	0	0	[]	
1	EdgeServer_2	0	2	[0, 2]	True	8	16384	131072	0	0	0	0	[]	
2	EdgeServer_3	0	3	[6, 0]	True	8	8192	131072	0	0	0	0	[]	
3	EdgeServer_4	0	4	[1, 3]	True	8	8192	131072	0	0	0	0	[]	
4	EdgeServer_5	0	5	[7, 1]	True	12	16384	131072	1	1024	1017	0	[]	
5	EdgeServer_6	0	6	[6, 2]	True	12	16384	131072	0	0	0	0	[]	
6	EdgeServer_1	1	1	[0, 0]	True	8	16384	131072	1	2048	53	1	[]	

4.4 Placement Algorithm

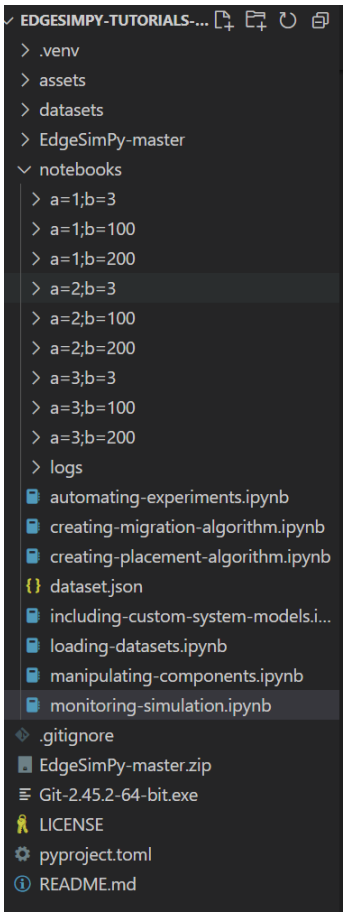


4.5 Migration Algorithm



5 Results from the proposed Code

Directory Structure for the running code with the parametric values a and b in the notebook directory.



5.1 Resuts:

```
logs = pd.DataFrame(simulator.agent_metrics["NetworkFlow"])
logs.describe(
)
```

[11]

	Time Step	Instance ID	Start	End	Source	Target	Actual Bandwidth	Data to Transfer
count	42.000000	42.000000	42.0	26.000000	42.0	42.000000	42.000000	42.000000
mean	3.500000	4.000000	1.0	2.615385	5.0	2.714286	6.428571	4.916667
std	1.728527	2.024243	0.0	1.626700	0.0	1.686181	3.857320	7.548959
min	1.000000	1.000000	1.0	1.000000	5.0	1.000000	2.000000	0.000000
25%	2.000000	2.000000	1.0	1.000000	5.0	1.000000	2.000000	0.000000
50%	3.500000	4.000000	1.0	3.000000	5.0	2.000000	6.250000	0.000000
75%	5.000000	6.000000	1.0	4.000000	5.0	4.000000	6.250000	10.000000
max	6.000000	7.000000	1.0	5.000000	5.0	6.000000	12.500000	23.750000

	Object	Time Step	Instance ID	Coordinates	Available	CPU	RAM	Disk	CPU Demand	RAM Demand	Disk Demand	Ongoing Migrations	Services	Registries	Layers	Images
0	EdgeServer_1	0	1	[0, 0]	True	8	16384	131072	0	0	0	0	[]	[]	[]	[]
1	EdgeServer_2	0	2	[0, 2]	True	8	16384	131072	0	0	0	0	[]	[]	[]	[]
2	EdgeServer_3	0	3	[6, 0]	True	8	8192	131072	0	0	0	0	[]	[]	[]	[]
3	EdgeServer_4	0	4	[1, 3]	True	8	8192	131072	0	0	0	0	[]	[]	[]	[]
4	EdgeServer_5	0	5	[7, 1]	True	12	16384	131072	1	1024	1017	0	[]	[1]	[ADD file:5d673d25da3a14ce1f6cf, /bin/sh -c se...	[registry, alpine, nginx, ubuntu, python, redi...
5	EdgeServer_6	0	6	[6, 2]	True	12	16384	131072	0	0	0	0	[]	[]	[]	[]
6	EdgeServer_1	1	1	[0, 0]	True	8	16384	131072	1	2048	53	1	[]	[]	[]	[] file:96
7	EdgeServer_2	1	2	[0, 2]	True	8	16384	131072	1	2048	53	1	[]	[]	[]	[] file:96
8	EdgeServer_3	1	3	[6, 0]	True	8	8192	131072	1	2048	2	1	[]	[]	[ADD file:5d673d25da3a14ce1f6cf]	[]
9	EdgeServer_4	1	4	[1, 3]	True	8	8192	131072	1	2048	2	1	[]	[]	[ADD file:5d673d25da3a14ce1f6cf]	[]
															[ADD	[registry, alpine, nginx,

6 Reference

[1]. Roman, R., Lopez, J., Mambo, M. (2018). Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, 78, 680–698.

[2]. Shahzadi, S., Iqbal, M., Dagiuklas, T., Qayyum, Z. U. (2017). Multi-access edge computing: open issues, challenges and future perspectives. Journal of Cloud Computing, 6, 1–13.

[3]. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19, 2322–2358.

[4]. M. Satyanarayanan, G. Klas, M. Silva, S. Mangiante, The seminal role of edge-native applications, in: 2019 IEEE International Conference on Edge Computing, IEEE, 2019, pp. 33–40.

[5]. M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vmbased cloudlets in mobile computing, IEEE Pervasive Computing 8 1125 (2009) 14–23.

[6]. H. Zhao, S. Deng, Z. Liu, J. Yin, S. Dustdar, Distributed redundancy scheduling for microservice-based applications at the edge, IEEE Transactions on Services Computing (2020).

[7]. J. Wang, Z. Feng, S. George, R. Iyengar, P. Pillai, M. Satyanarayanan, 1130 Towards scalable edge-native applications, in: ACM/IEEE Symposium on Edge Computing, 2019, pp. 152–165.

[8]. P. Souza, A. Crestani, T. Ferreto, F. Rossi, Latency-aware privacypreserving service migration in federated edges, in: International Conference on Cloud Computing and Services Science, 2022, pp. 288–295. 1135
[9]. P. Souza, T. Ferreto, F. Rossi, R. Calheiros, Location-aware maintenance strategies for edge computing infrastructures, IEEE Communications Letters 26 (2022) 848–852.
[10]. Yu, Z., Xu, X. and Zhou, W. (2022). Task Offloading and Resource Allocation Strategy Based on Deep Learning for Mobile Edge Computing. *Computational Intelligence and Neuroscience*, 2022, pp.1–11. doi:https://doi.org/10.1155/2022/1427219.
[11]. Vaisman, R., Kroese, D.P. and Gertsbakh, I.B. (2016). Improved Sampling Plans for Combinatorial Invariants of Coherent Systems. *IEEE Transactions on Reliability*, 65(1), pp.410–424. doi:https://doi.org/10.1109/tr.2015.2446471.
[12]. Li, X. (2021). A Computing Offloading Resource Allocation Scheme Using Deep Reinforcement Learning in Mobile Edge Computing Systems. *Journal of Grid Computing*, 19(3). doi:https://doi.org/10.1007/s10723-021-09568-w.
[13]. Ullah, I., Lim, H.-K., Seok, Y.-J. and Han, Y.-H. (2023). Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach. *Journal of Cloud Computing*, 12(1). doi:https://doi.org/10.1186/s13677-023-00461-3.
[14]. Mourita Mozib (2023). Distributed Deep Learning Based Framework to Optimize Real-Time Offloading in Mobile Edge Computing Networks. *International journal of science and research*, 12(6), pp.1812–1827. doi:https://doi.org/10.21275/sr23603125305.
[15]. Yang, H., Xu, C., Liu, S., & Zhang, J. (2020). A Privacy-Preserving Task Offloading Scheme for Edge Computing in the Internet of Things. IEEE Internet of Things Journal, 7(7), 6124-6136.
[16]. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. & Ayyash, M. (2015), 'Internet of things: A survey on enabling technologies, protocols, and applications', IEEE communications surveys & tutorials 17(4), 2347–2376.
[17]. Alghamdi, I. (2021), Computation offloading in mobile edge computing: an optimal stopping theory approach, PhD thesis, University of Glasgow.
[18]. Cardellini, V., De Nitto Person´e, V., Di Valerio, V., Facchinei, F., Grassi, V., Lo Presti, F. & Piccialli, V. (2016), 'A game-theoretic approach to computation offloading in mobile cloud computing', Mathematical Programming 157, 421–449. Hu, Y. C.,
[19]. Patel, M., Sabella, D., Sprecher, N. & Young, V. (2015), 'Mobile edge computing—a key technology towards 5g', ETSI white paper 11(11), 1–16. Liu, L., Z. Y. . S. L. (2019),
[20]. 'Task offloading and resource al location for mobile edge computing by deep reinforcement learning based on sarsa', Ieee Access . Mao, Y., Zhang, J., Song, S. & Letaief, K. B. (2017),
[21]. 'Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems', IEEE Transactions on Wireless Communications 16(9), 5994–6009. Rego, P. A.,
[22]. Trinta, F. A., Hasan, M. Z., de Souza, J. N. et al. (2019), 'Enhancing offloading systems with smart decisions, adaptive monitoring, and mobility support', Wireless Communications and Mobile Computing 2019.
[23]. Satyanarayanan, M., Bahl, P., Caceres, R. & Davies, N. (2009), 'The case for vm-based cloudlets in mobile computing', IEEE pervasive Computing 8(4), 14–23.
[24]. Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J. & Wang, W. (2017), 'A survey on mobile edge networks: Convergence of computing, caching and communications', Ieee Access 5, 6757–6779.
[25]. Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S. & Zhang, Y. (2016), 'Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks', IEEE access 4, 5896–5907.
[26]. Souza, P.S., Ferreto, T. and Calheiros, R.N. (2023). EdgeSimPy: Python-based modeling and simulation of edge computing resource management policies. Future Generation Computer Systems, [online] 148, pp.446–459. doi:https://doi.org/10.1016/j.future.2023.06.013.