COMP 40660: Advances in Wireless Networking

School of Computer Science, University College Dublin, Ireland

Spring 2023

Assignment 2: Edge Computing (Group project)

This assignment is worth 15% of the overall grade.

Multi-Access Edge Computing (MEC) and Fog computing are the leading edge computing paradigms proposed to realize the guaranteed service requirements of 5G. Edge computing paradigms offer a storage and processing infrastructure at the edge of the mobile network (preferably at a mobile base station) to facilitate ultra-low latency and higher bandwidths for heterogeneous services catered by IoT deployments. Virtualization is the key technology considered for implementing versatile infrastructure at the edge. On the same note, lightweight virtualization presents unique features that improve the feasibility of the autonomous edge computing concept.

Mobile offloading, or computational offloading is the process of outsourcing heavy computing tasks to the edge environment, which are unfeasible to perform in the UE or the mobile device. A typical offloading task is different from a data offloading scenario. The raw data to be processed is conveyed to the MEC edge, while the ultimate decision, value, or classification is notified to the UE after the computation. The main benefits of mobile offloading are the extension of the battery life of UEs, the ability of users to employ sophisticated services that demand beyond their UE specifications, and improved capacity for storage within the edge infrastructure.

In this assignment, the intended outcomes are: The student,

- to understand the significance of lightweight virtualization for edge computing
- to learn basic knowledge of docker virtualization technology
- to develop a dockerized environment emulating an edge computing platform
- to establish an edge-to-edge connectivity between two docker containers
- to develop a mobile offloading scenario in the established edge platform

The students are expected to perform the following two tasks in this assignment.

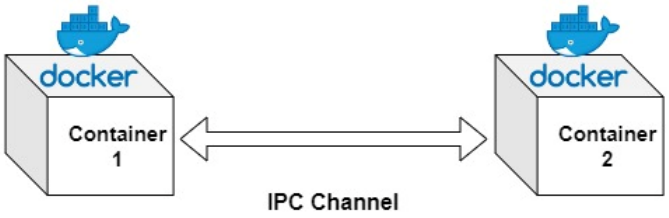
Task 01: Basic Docker Virtualization (6 Marks)

- Create a dockerized environment in your computer/ PC (Linux environment preferred. Can be hosted inside a VM) and run at least 3 well-known docker images from the docker hub (i.e., hello-world, BusyBox, Nginx, Redis, Alpine,…etc.).
- Show the running docker containers in your machine. Remove all the running containers and their images and show the list of images in the virtual domain.
- Follow the instructions at https://www.docker.com/blog/how-to-use-the-official-nginx-docker-image/ to run a basic Nginx web server in your PC/VM. Add a custom HTML message with your Group members' names and run the server. Show the outcomes.
- Boot up a basic ubuntu container. Install any Linux package inside the container, such as nano or IP-Utils.
- Create a directory with your group name inside the container. Inside the container, create a text file and include the group members' names in it. Show the created directory and the file content.
- Commit the Ubuntu container and push its image to the docker hub. Mention the link to the exported docker hub image.

Task 02: Docker Networking (6 Marks)

https://docs.docker.com/network/

- Follow the instructions at https://docs.docker.com/network/network-tutorial-standalone/ use a default bridge to connect three alpine containers and run ping commands to test their connectivity. Show the ping results.



- Establish an Inter-Process Communication (IPC) channel between two Ubuntu containers following the instructions at https://medium.com/techanic/docker-containers-ipc-using-sockets-part-1-2ee90885602c
- Leveraging the created IPC channel, send parametric values (at least 50) from container 1 to container 2, emulating an offloading scenario; compute the mean, median, and standard deviation of these parameters at container 2. You might have to extend the python script to establish the IPC communication. Send the computed stats to container 1 and display all the results at each container.
- Emulate another offloading scenario that is relevant for future applications.

Deliverables

a) Design document describing the design of your system, observations, and performance statistics (offloading time, etc.) (Maximum 8 pages); (3 Marks)

b) A video (Max. 6 min and 500 MB) on the presentation of the use of your system (especially the docker networking part and the offloading part);

c) Links to the created Docker images (Docker Hub or GitHub) with a clear README file.

The evaluation of Task 01 and Task 02 will depend on the b) and c) deliverables.

This assignment is a group project. You can form a group of up to 3 people of your own choice. Individual and groups of 2 are also allowed. When submitting to Brightspace, the Design document (.pdf / .doc / ...etc.), recording of the presentation (.mp4), and the README file should be submitted as separate files. Only one submission per group is sufficient. Only the last submission will be graded.

Please follow these instructions to complete the submission successfully.

Submission deadline: 11.59PM on 30th April 2023

End of the assignment
