Software Configurations in Edge Computing
Carlos Colon-Barrios, Dr. Bozorgzadeh 2, Tootiya Giyahchi 2
NSF REU IoT-SITY
1 University of Puerto Rico at Arecibo 2 University of California, Irvine

OBJECTIVE
Show the Properties of Edge Computing Software

INTRODUCTION
- Microsoft and Amazon both offer software architectures that facilitate communication between the end and edge devices.
- A Raspberry Pi 3 B will be used as and end device, which will connect to the Amazon and Microsoft cloud service.
- Microsoft Azure will be utilized to simulate sensor data processed by the Raspberry Pi and send it to the Azure cloud.
- Amazon Greengrass will be used to send basic Lambda functions that are processed through the Raspberry Pi to be sent to the AWS cloud.

PROJECT OVERVIEW
This project is an introduction to the services provided by Amazon and Microsoft. The purpose of this project was to eventually integrate edge computing into any edge device, to reduce workload from the limited processing the Raspberry Pi offers.

The edge computing architecture facilitates processing high workloads given by deep learning, machine learning techniques like image recognition.

- The architecture on how Azure’s hub and Greengrass’ core are shown below.

RESULTS
The Raspberry Pi 3 B model was used to download AWS Greengrass and Microsoft Azure dependencies and run basic functions.

The Raspberry Pi was only able to connect directly to the Azure cloud and run a C program that simulated temperature and humidity information as well as a hello world.

On AWS Greengrass, a Lambda function written in Python was made. The hello world on Greengrass is much similar in theory, but applying simple functions was much easier and faster through Azure. Both record messages sent by the raspberry pi at their cloud and have the option to integrate machine learning with image recognition for future work.

ACKNOWLEDGEMENTS
Dr. Nalini, Dr. Sharnnia Artis and Tootiya Giyahchi

FUTURE WORK
- Configure the Pi to be able to successfully send most workload to the laptop and back to compare efficiency versus sending it directly to cloud.
- Do this is a much larger scale such as:
  1. Having a Raspberry Pi in a car with camera sensors and test the response time of cloud and edge computing.
  2. Implementing a self-checkout using the camera for the visually impaired.

METHODOLOGY
1. Install Amazon Greengrass on raspberry pi
2. Install Microsoft Azure dependencies on raspberry pi
3. Study user interface, accessibility and basic programs
4. Compare results

This methodology is to learn the basics of the software provided by Amazon and Microsoft. Also, after these steps are followed, we can determine how easy or difficult is interacting with both software architectures and the features given when configuring an end device to their cloud.

LIMITATIONS
- To run AWS’s python dependencies the Raspberry Pi has to be up to date, but to use a machine learning application called Mxnet, it would only run on older versions of the Raspbian OS.
- At first Microsoft’s edge computing platform, Azure edge, was going to be used but had more limitations, one being my laptop not having a paid Windows Pro version that would’ve allowed me to use my laptop as an edge device.
- Not having a camera for Pi, so we were only using simulated information.

SOFTWARE FRAMEWORK OF BOTH SERVICES

Microsoft and Amazon both offer software architectures that facilitate communication between the end and edge devices.