Edge Computing Platforms

October 21^{st}, 2019

1 Project Description

The following is a comparison of the edge computing services offered by the current leaders in the cloud computing industry, Amazon and Microsoft. This study will look to compare the services, see where they excel at, and potential areas for improvement. For Amazon, the main tools used are AWS IoT Greengrass and AWS Lambda Edge. For Microsoft, the primary tool is Azure IoT Edge. Here there will be an experiment with both platforms to measure specific areas such as latency, security, data caching, and overall performance. Applications will be run from multiple mobile devices connected to the cloud network with numerous tests in place. Each test will be documented and external factors such as fees and ease of use will be noted along the process.

AWS IoT Greengrass allows cloud functionalities to operate on sets of devices. Greengrass core is downloaded onto the device which establishes a group. Following this other devices can join the group. From here the AWS IoT Greengrass is programmed to filter specific data from the device and transfer necessary information to the cloud. This benefits of this include: responsiveness to local events as they happen, offline operations, and high security features. It should be noted that latency with this will depend heavily on the environment and connection.

There is an AWS IoT device tester which is an automation tool for linux-based IoT devices which will determine whether or not the device will run AWS IoT Greengrass and operate with the services. Greengrass tests use the CLI to execute test cases on the device of choice and the results are documented on the computer. AWS Lambda Edge takes advantage of the large network of edge locations to provide local, low latency compute power. This allows you to run code closer to users in order to improve importance and reduce latency.

Azure IoT Edge allows you to move workloads to run on IoT devices on the edge of the network through standard containers. The result is that the device reduces the amount of time spent communicating with the cloud and reacts swiftly to local changes and operates efficiently in offline periods.

2 Timeline

Date	Description
October 30	Experiment evaluating compute at local, edge, and cloud
November 15	Implement algorithm to compute on edge (AWS and Azure)
November 25	Experiment evaluating results for both platforms
December 3	Complete presentation slides
December 10	Complete final report height

Table 1: Project timeline

References

[1] Eswara, V., Srivastava, G., & Biswas, S. (2017, June). RIOTNet: Reactive IoT Control Network. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 643-650). IEEE.

[2] Kurniawan, A. (2018). *Learning AWS IoT: Effectively manage connected devices on the AWS cloud using services such as AWS Greengrass, AWS button, predictive analytics and machine learning*. Packt Publishing Ltd.

[3] Noghabi, S. A., Kolb, J., Bodik, P., & Cuervo, E. (2018). Steel: Simplified development and deployment of edge-cloud applications. In *10th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 18)*.