CIS-490H Edge Computing

With Dr. Zheng Song

Paper Review: Week 6

“Pushing Serverless to the Edge with WebAssembly Runtime”

Student: Demetrius Johnson

08 February 2023

1. Summary.

(1) Motivation

The primary motivation for this paper is to promote a possible solution to help solve the cold start problem in local edge server nodes, which is currently addressed by methods that go against the *scale-to-zero* principle. They invoke the newest WebAssembly interface and apply it so that it can replace the classic Docker container method which incurs too slow of cold-start times. They call it the WOW (WebAssembly Container Runtime for OpenWhisk), and it serves as a viable solution to providing Function as a Service (FaaS) to be ran on fog/edge nodes, while having the advantage of being able to be ran across non-homogenous nodes, since WebAssembly is about code that is contained securely inside of the browser memory that can be executed safely, and compiled to various levels that can make it near native run time on a machine.

(2) Contribution

The research provides the information of the infrastructure (environment) required to execute Wasm serverless functions, developing the WOW framework which make it possible to execute Wasm functions in a server running Apache OpenWhisk. They provide the experimental results of the WOW system under various conditions that can be replicated and the information used to decide if and by how much deploying this serverless solution could improve an IoT application that relies on edge computing. The code for the system is opensource, and the contributions they made make it possible for anyone developing an edge computing framework to test an IoT application and compare it to traditional FaaS solutions.

(3) Methodology and/or argument

First, they define workload types that they will test on their WOW platform: CPU and I/O bound workloads. They give various outputs of handling the workloads type in scenarios where one or both workloads are present in the system. They deploy their system on Raspberry Pis, and they use more Pis to deploy a traditional system that they will compare their system to. They collect metrics primarily about cold start times and runtimes to measure latencies of running the various scenarios under given workload combinations.

(4) Conclusion

Overall, the results of their experiment provide substantial evidence that the current Docker solution may be an obsolete edge computing platform choice when it comes to running FaaS for IoT applications. Various compilation levels of Wasm proved to be much faster and more memory efficient versus using the traditional Docker method to run FaaS on edge devices that have limited computational power and resources. Docker replaced Virtual machines as lightweight, and now this Webassmebly execution environment method may prove to be an even lighter-weight solution to for providing a FaaS platform.

2. Critique.

I would say this paper is heavy duty on a lot of terminology about a relatively new field. Although for the most part they do describe much of the abbreviations and terminology, I think it would be wise to have a glossary of terms of some sort to make it easy for a reader to reference all of these relatively new (and heavily abbreviated) concepts that all fit together to prove a point. In fact, many papers should include a glossary of terms that specific lists very key vocabulary and provide the definition based on the context of their paper.

Also, their functional diagrams lacked details to truly show an overview and detailed view of the full system. They need more diagrams, including a diagram of traditional FaaS platform versus their WoW platform so it is easier for a reader to understand the difference between the two systems.

3. Synthesis.

I would recommend that this paper be exactly replicated in a real environment to test its performance versus traditional docker methods and results collected for comparison. I also would recommend that they update this paper to include some of the missing features they mentioned under *Limitations and open* challenges section.