Optimizing Microservice-based Applications for Cloud and Edge Architectures

Undergraduate Thesis

by

Yan (Oscar) Yu

CS 4490Z
Thesis Supervisor: Hanan Lutfiyya
Course Instructor: Nazim Madhavi

Department of Computer Science
Western University, London, Ontario N6A 5B7, Canada
August 12, 2024

Abstract

Edge computing has been the subject of much attention in the software development space over the last several years as the limitations of traditional cloud computing models continue to be exposed by an increasing number of connected IoT and internet-enabled devices that require real-time computing. As this new computing paradigm becomes more prevalent in the industry, it is important that software is developed effectively to take advantage of the benefits that edge computing brings to the table.

In this paper, we attempt to establish an understanding of core principles that will enable the effective design and development of distributed software systems that can be easily deployed and optimized for various architectures of computing models – primarily hybrid cloud edge networks.

Contents

1 Introduction

2 Background and Related Work

This thesis is based on existing work in multiple fields, including

2.1 Cloud Computing

2.2 Edge Computing

2.3 Microservice Architecture

3 Research Objectives

(O1) Understand what applications may benefit from deployment to edge networks

(O2) Understand how to identify performance characteristics of individual components within a software application

(O3) Understand how applications can be developed to optimize for deployment across different network architectures

(O4) Understand how various hosting patterns can affect latency and user experience

4 Methodology

4.1 Application Development

Not all software is created equally, with a vast range of performance characteristics and behaviour across various applications when developed for differing use cases. For the purposes of demonstrating this behaviour in order to identify favourable conditions for edge deployments, we develop a minimal microservice-architecture application for real-time object recognition consisting of a web interface, API gateway/web server, and an inference service. (O1)

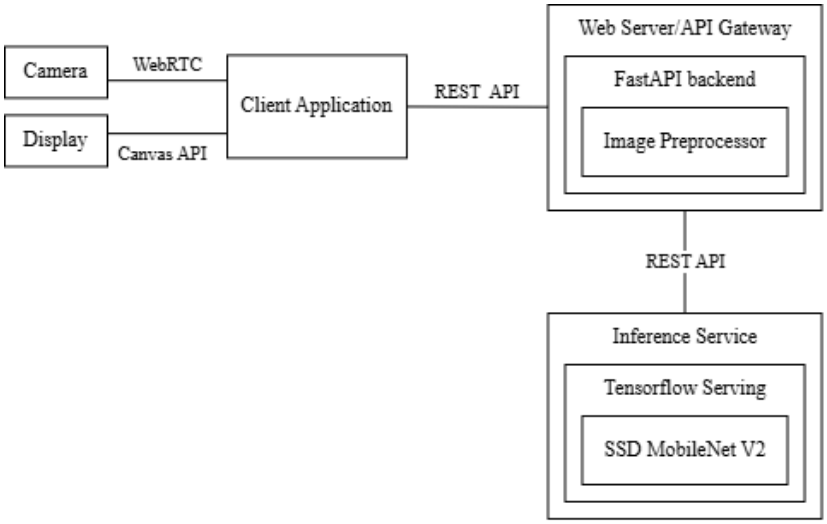


Figure 1: Example architecture

Real-time object recognition using a CNN like

4.2 Deployment Architecture(s)

To demonstrate the feasability of developing an application compatible with multiple network architectures and gather data regarding the performance of an identical application across multiple architectures, we

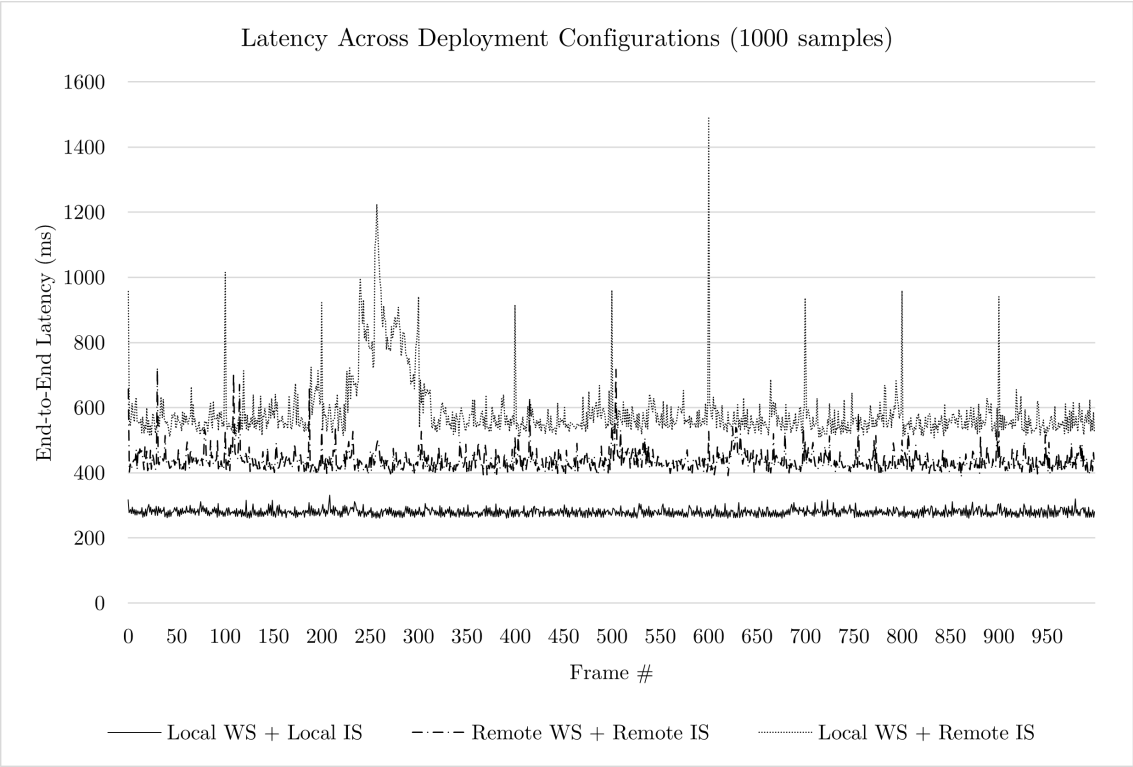
4.3 Performance Analytics

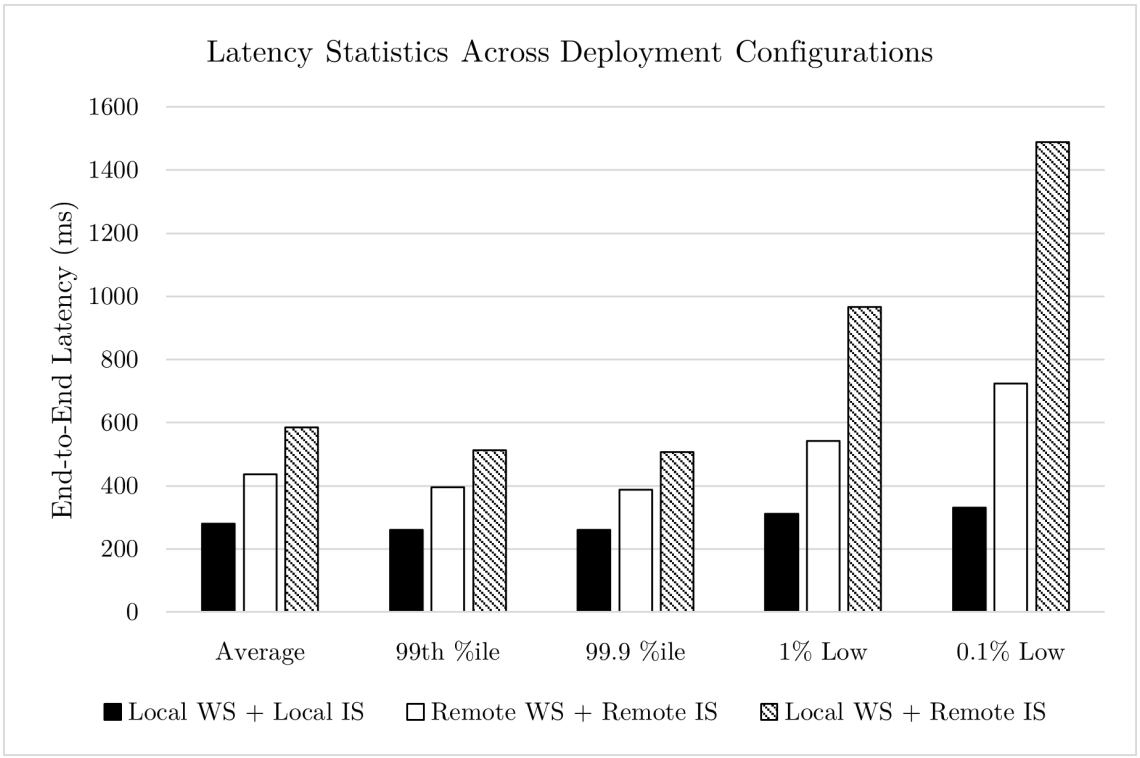
5 Results

5.1 Test Application

5.2 Data

The graphs below show the latency data collected from deploying the sample application across 3 different configurations of cloud and simulated edge compute servers





5.3 Latency Differences

6 Discussion

6.1 Implications

6.2 Limitations and Generalizations

- Hardware limitations - Limited model selection - Relatively simple system design - Lack of edge deployments - No service level latency monitoring - Lack of cloud hardware control - Only latency measurements, no hardware usage or identification of other bottlenecks

7 Conclusions and Future Work

- Perform more analysis with more performance analytics

8 Reference List