# CS 695 - Virtualization and Cloud Computing
# Project Report: Synergia
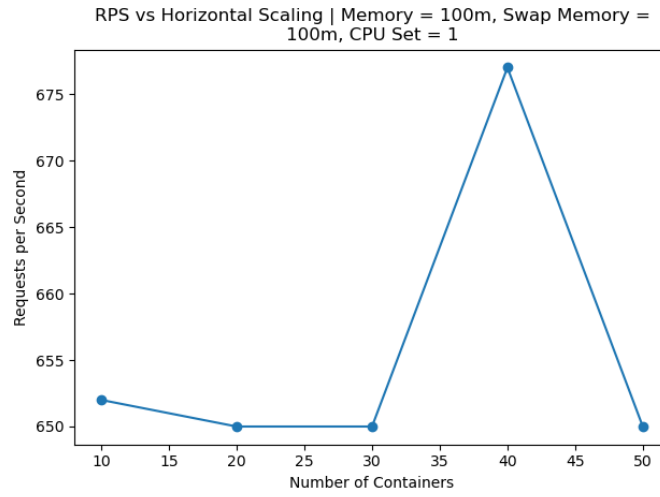
Abhijit Amrendra Kumar (210050002)

## 1  Introduction

The implementation of this project involved understanding and upgrading the code from Assignment 3.
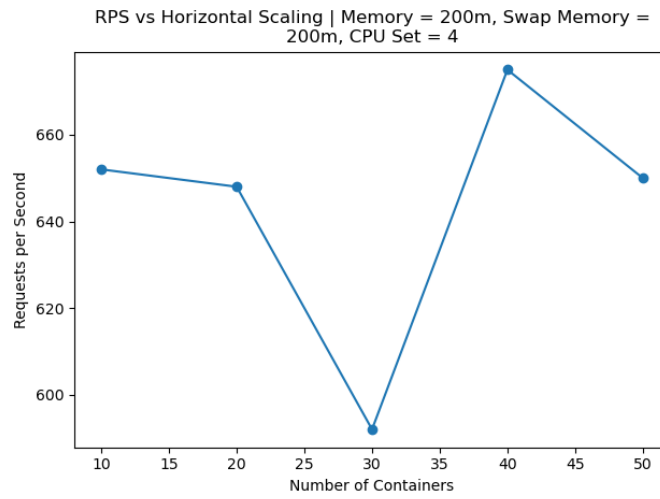
Here are the list of changes and upgrades implemented in this project.

- Replaced the custom-made container implementation with Docker containers.

- Added concurrency fixes (mutex) to the `counter-service` to allow multiple clients to access the service simultaneously without landing into race conditions.

- Modified the `external-service` python app to use `Quart` instead of Flask to allow for asynchronous processing of requests.

- Added a load balancer using NGINX to distribute the incoming requests to the multiple `external-service` instances.

- Added a scaling script to

    1. horizontally scale the `external-service` instances by increasing/decreasing the number of containers.
    2. vertically scale the `external-service` instances by increasing/decreasing the memory, swap memory and cpu cores for all containers.

- Used Apache ab for load testing (number of requests per second) over multiple configurations of the containers, and plotted the results.
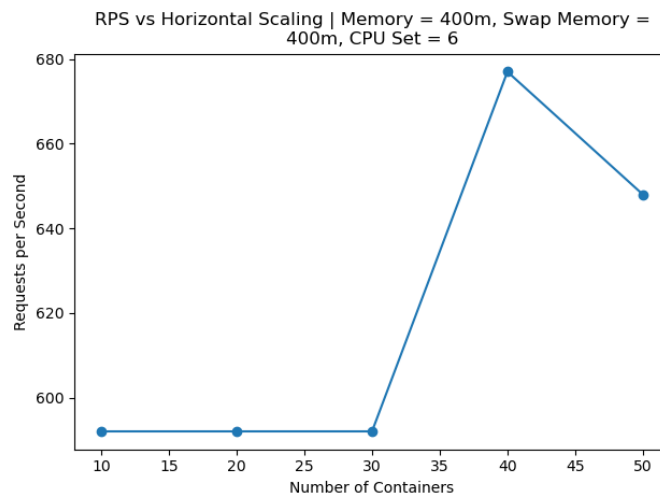
Because the individual programs do not exceed even the base RAM configuration, there is hardly any difference between the maxima and the minima of the 3 plots. For each individual plot, the generic trend is that the RPS (Requests Per Second) increases with horizontal scaling.

RPS vs Horizontal Scaling | Memory = 100m, Swap Memory = 100m, CPU Set = 1

(a) Vertical Scaling: 100m, 100m, 1



RPS vs Horizontal Scaling | Memory = 200m, Swap Memory = 200m, CPU Set = 4

(b) Vertical Scaling: 200m, 200m, 4



RPS vs Horizontal Scaling | Memory = 400m, Swap Memory = 400m, CPU Set = 6

(c) Vertical Scaling: 400m, 400m, 6