Assignment 1

FIT5225 2022 SM1

Cloudiod:

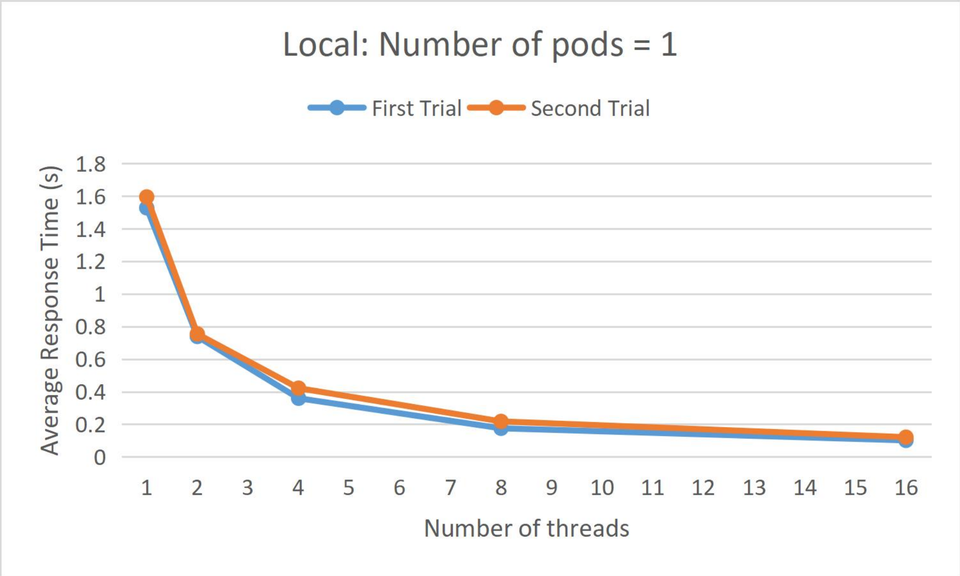
Creating and Deploying an Image Object Detection Web Service within a Containerised Environment in Clouds

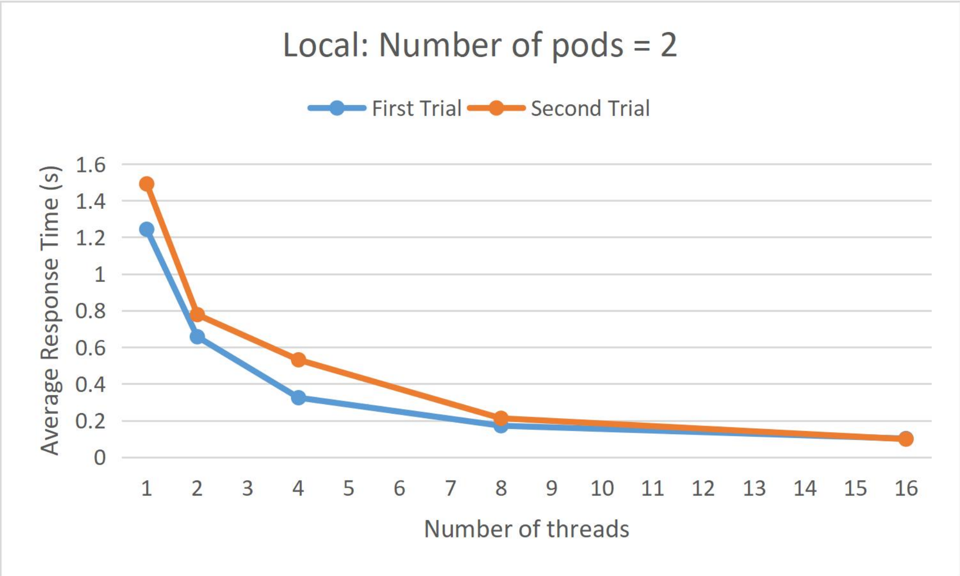
Full name: Zhexuan Hao

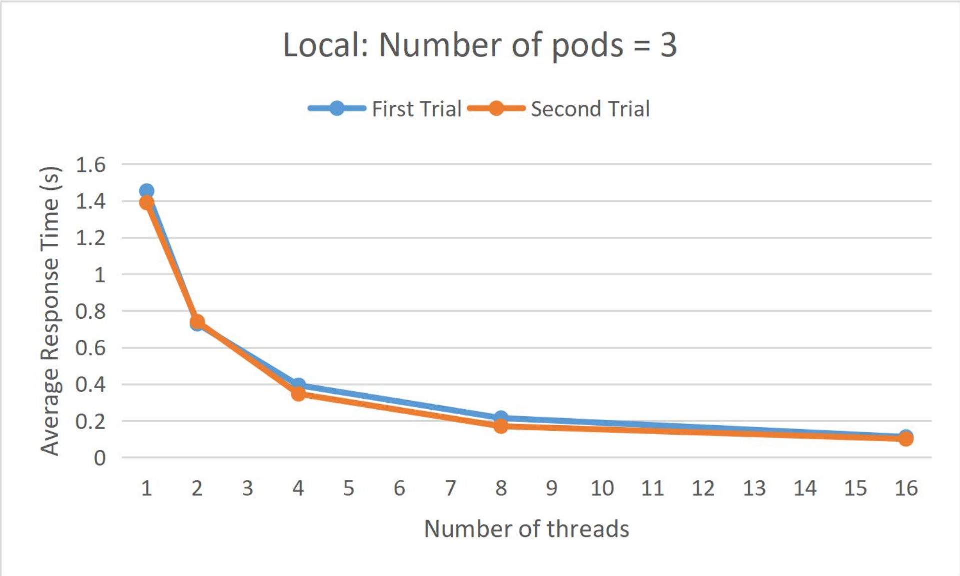
Tutor name: Peibo Duan

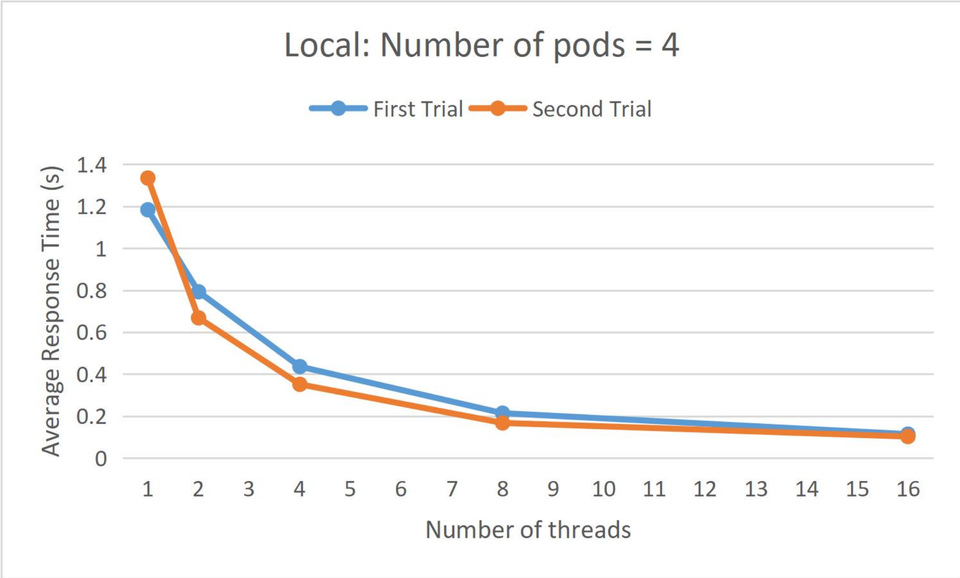
Student number: 33414351

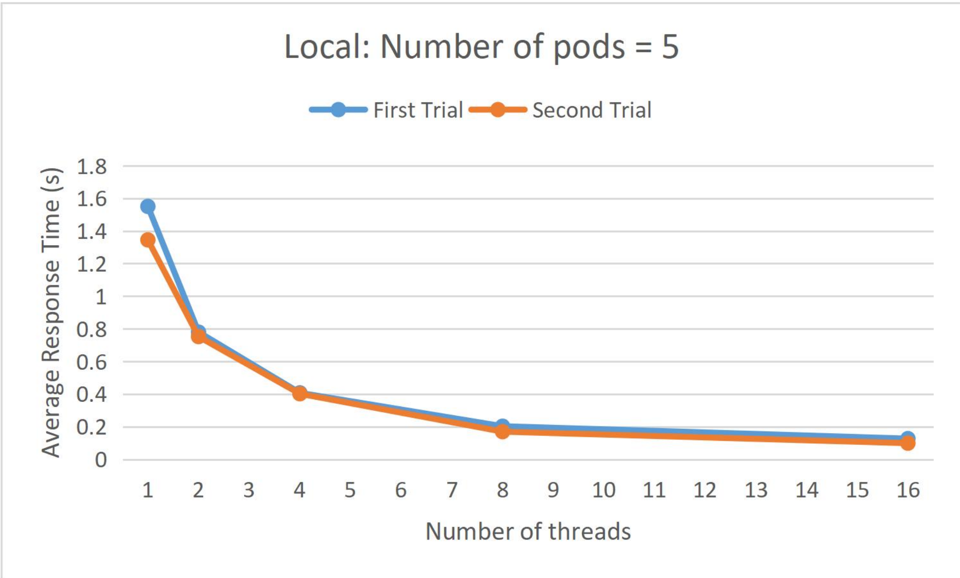
- A plot showing the average response time of the web service versus the number of pods for different number of threads for the local client.



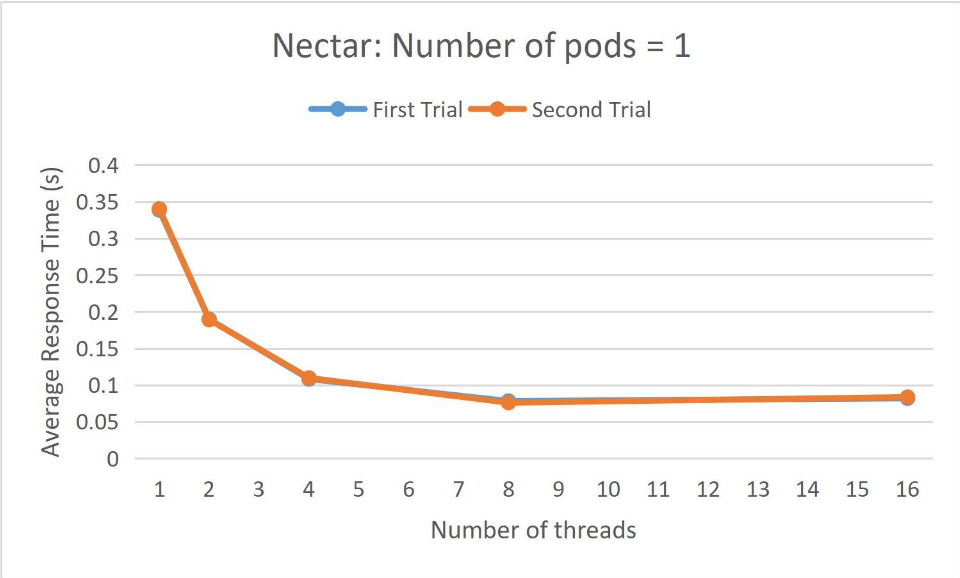


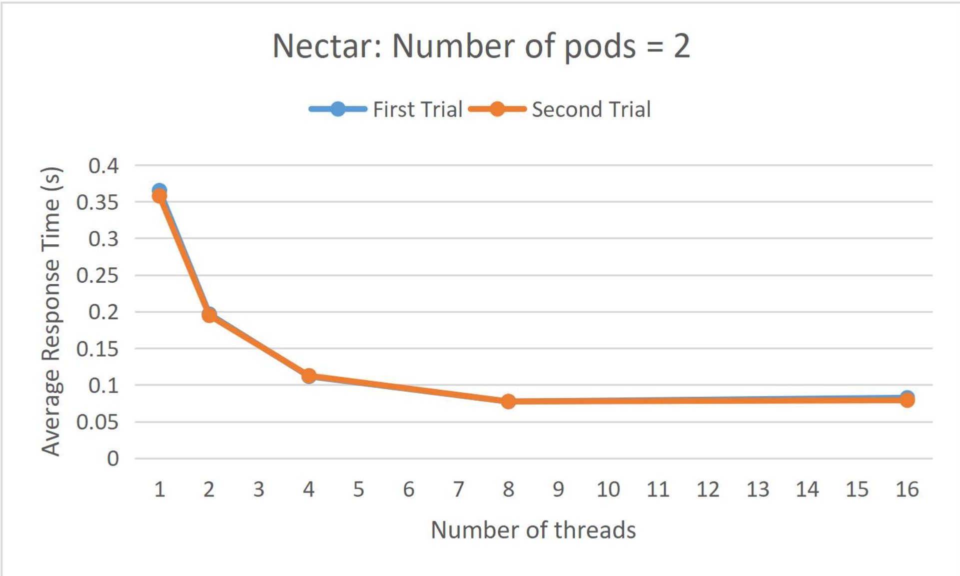


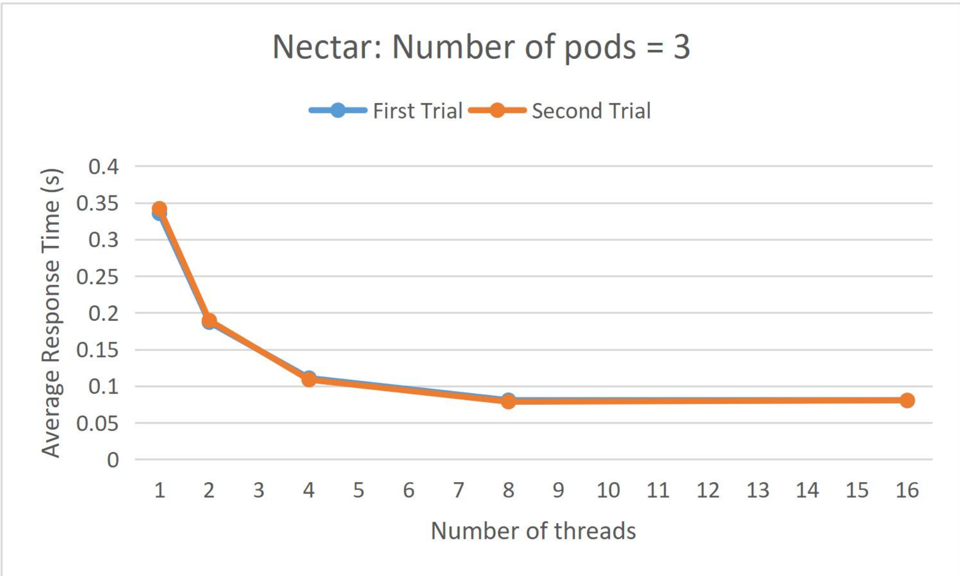

Local: Number of pods = 4
First Trial
Second Trial
Average Response Time (s)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Number of threads

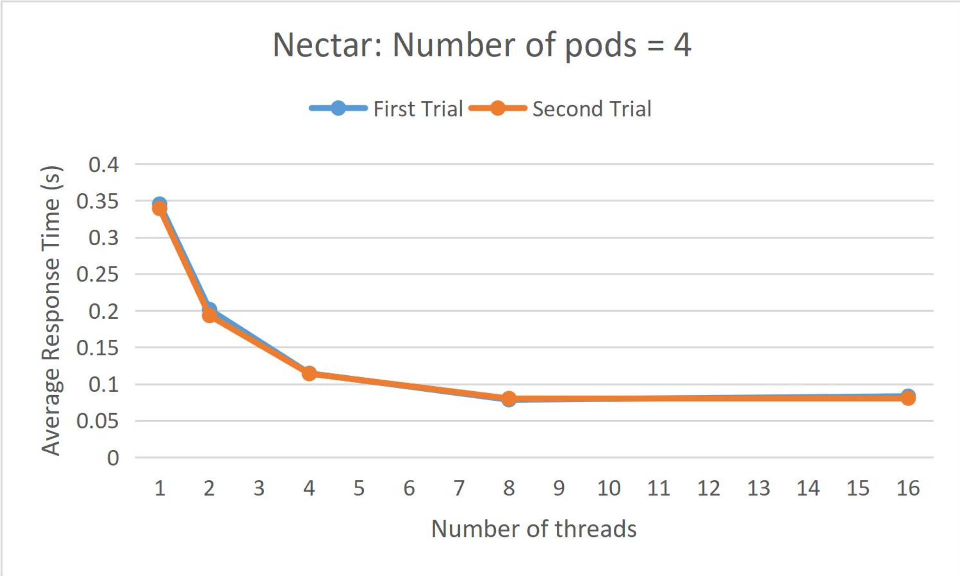

Local: Number of pods = 5
First Trial
Second Trial
Average Response Time (s)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Number of threads

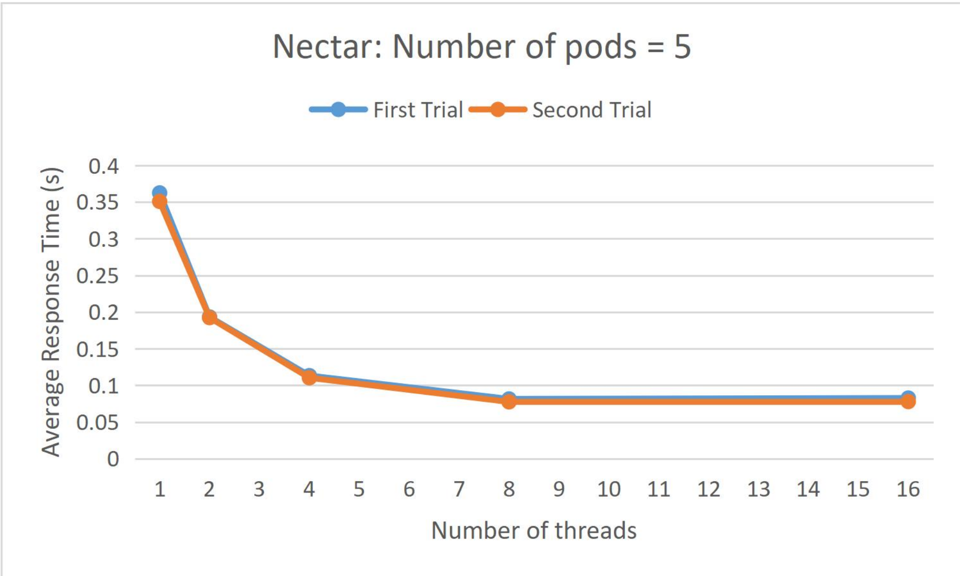
- A plot showing the average response time of the web service versus the number of pods for different number of threads for the Nectar client.








Nectar: Number of pods = 4
First Trial
Second Trial
Average Response Time (s)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Number of threads


Nectar: Number of pods = 5
First Trial
Second Trial
Average Response Time (s)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Number of threads

- Explain and justify results, plots, trends and observations in your experiments.

As can be seen from the above data and line chart, when the same server has a fixed number of Pods, the number of threads is inversely proportional to the running time. The more threads, the shorter the run time, but only if the number of threads is not too large. On the Nectar client, the number of threads varies from 8 to 16 and the running time remains the same. The reason may be that there is a bottleneck in the operating system. Too many threads will waste a lot of computing and resources on threads management, because threads need to allocate resources and memory. So too many threads will be counterproductive.

However, when the number of threads remains the same, the change in the number of pods has little effect on the running time, for the following reasons: In this experiment, only one container is loaded into a pod, and pod is the smallest unit called, so even if the number of pods changes, only one container is in use, which has no effect on the speed.

As for the differences in the running time between the different clients, it can be seen from the figure that the Nectar client takes up much less time than the local client, which may be related to the installed operating system, network, distance, and operating environment. It is difficult to analyze the specific variables.

- Select three challenges of your choice from the list of distributed systems challenges discussed in the first week lecture, give a practical example from your project that illustrates that challenge and how it is addressed in your system (500 words).

Distributed systems face many challenges such as heterogeneity, openness, security, scalability, failure handling, concurrency, transparency and quality of service. One of the challenges that we faced while creating and deploying YOLOv3 on Kubernetes and Docker was scalability.

Scalability is the ability of a system to handle an increasing amount of work by adding resources to the system. In a controlled experiment for our project, we needed to handle more and more requests by increasing the number of threads without crashing or slowing down. We address this requirement by using Kubernetes, which consists of one master and two nodes with 4 core 8G of RAM, limiting resources per deployment to 0.5 core and 512MiB, and processing requests is significantly faster after increasing the number of threads, which can be adjusted as required. It can handle a large number of requests and reduce resource consumption when demand is low.

Another challenge we faced was failure handling. Failure handling refers to the ability of a system to continue operating in the event of one or more failures. In our project, we had to make sure that our system could handle failures, such as node failures or container failures, without crashing or losing data. During the experiment, there was an occasional server crash where node2 was unavailable and Pod could not be assigned to it, but node1 was still running because Kubenetes has built-in fault tolerance mechanisms such as automatic container restart and node failover. The pods of node2 are assigned to node1. We solved this challenge by using Kubernetes.

Finally, another challenge we face is security. Security refers to the ability of a system to prevent unauthorized access and data leakage. In our project, we have to ensure that our systems are secure and that data is protected at all times. We solve this problem by using Docker, which provides container isolation and security features such as namespaces. We place images and containers in our own namespace so that the rest of the sections are not affected.