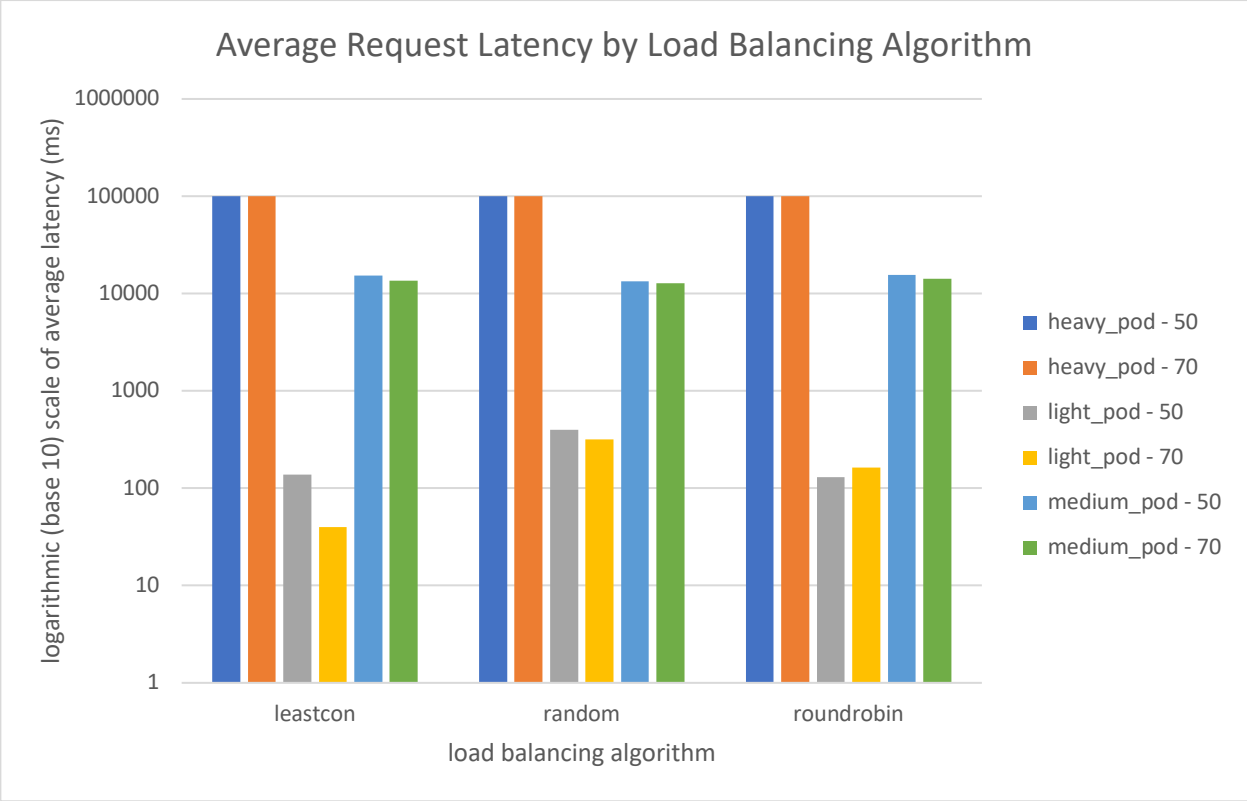
In this assignment, a simple cloud system with a load balancer (HAProxy) was implemented by Group 3. In particular, the effects of various load balancing algorithms and node configurations on the average throughput and latency per pod were investigated.

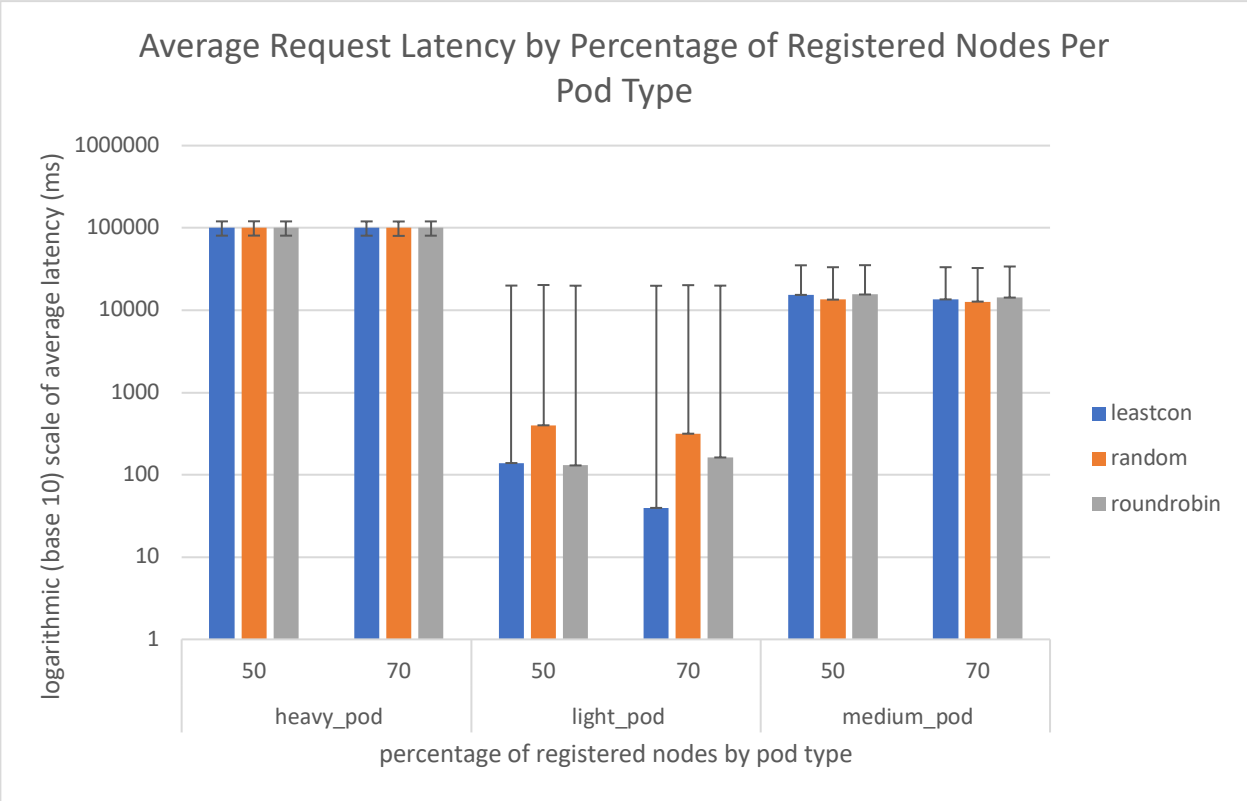
A script was created to set up the cloud with node configurations of 50% and 70% of the maximum number of nodes online per pod, to inject multiple requests at a rate of 5 requests per second, and to record the request latency for each of these requests. This script was run for each of the load balancing algorithms: round robin, random, and least connected server. It was also run for each of the pod types: light, medium, heavy. The light job had light computations on small data: random string generation. The medium job had medium computation on medium data: image transformation. The heavy job had heavy computations on large data: video transformations.

The results are presented below:

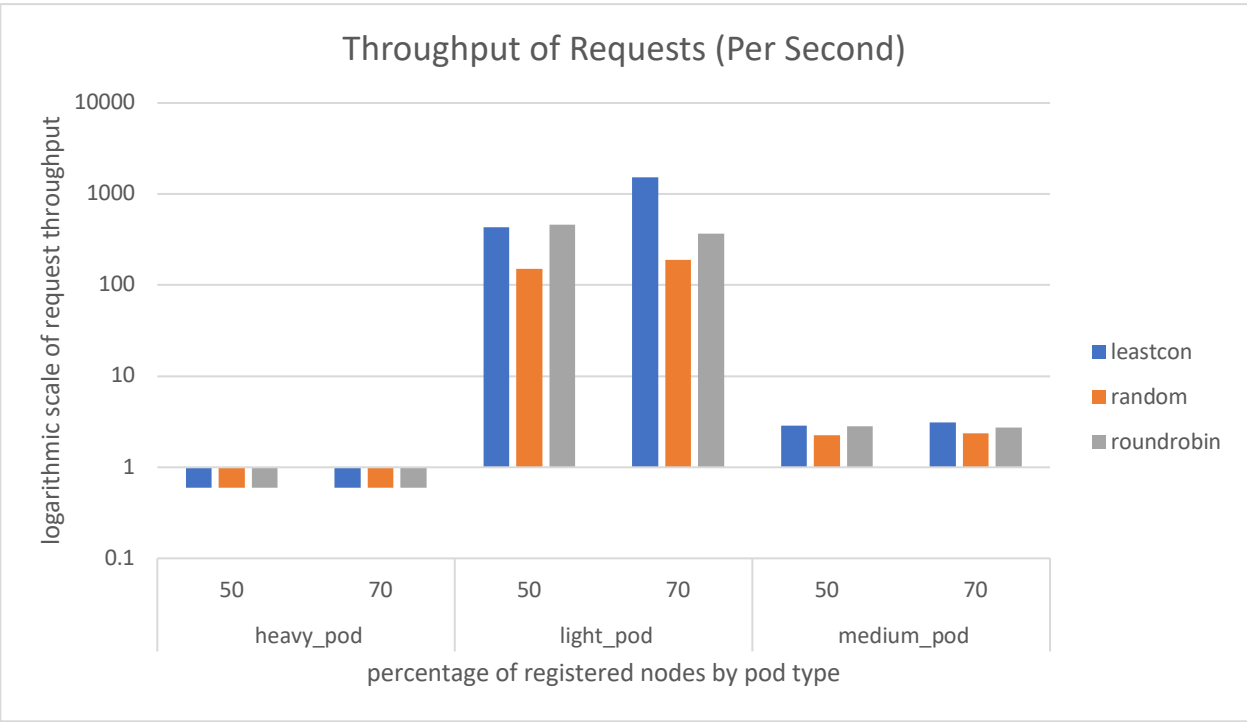


Across all three load balancing algorithm types, the latency goes from lowest to highest for light, medium, and heavy pods, as expected due to the nature of their jobs. There is not a large difference between the different algorithms' performance except for the light pod's requests in which the least connected server algorithm outperformed round robin and random. Generally, the node 50% node configuration had a higher average latency, as expected.

The figure below presents the same data in another representation with its standard error.



The average throughput of requests for each node configuration per pod was also measured and is shown below. The throughput is highest for the light pod, and lowest for the heavy pod, which can be explained by their jobs having a large difference in computational duration.



Bringing the latency and throughput together: the least connected server algorithm improves throughput but not latency, and higher availability slightly improves throughput and latency. As a result of these tests, we can conclude that the least connected server algorithm may be the best choice for our simple cloud implementation.