Experiment Report

In this experiment, we measured the average throughput and latency per pod by changing the number of nodes ONLINE per pod (50%, 70%) and using different load balancing algorithms (round robin, random, least connected server). We used a bash script to send requests to the server and measure the results.

Setup

We are using the following bash script to test the load balancer:

```bash
#!/bin/bash

url="localhost:5002/heavy"
success_count=0
total_count=0

start_time=$(date +%s)

# Send 10 requests in parallel using xargs
seq 1 100 | xargs -n1 -P100 -I{} sh -c 'curl -s "$0" >/dev/null 2>&1 && echo "success"' $url | wc -l > /dev/null
success_count=$?

end_time=$(date +%s)
duration=$((end_time - start_time))

echo "Sent 10 requests in parallel."
echo "It took $duration seconds to send and complete all requests."
echo "Out of $total_count requests, $success_count were successful."
```

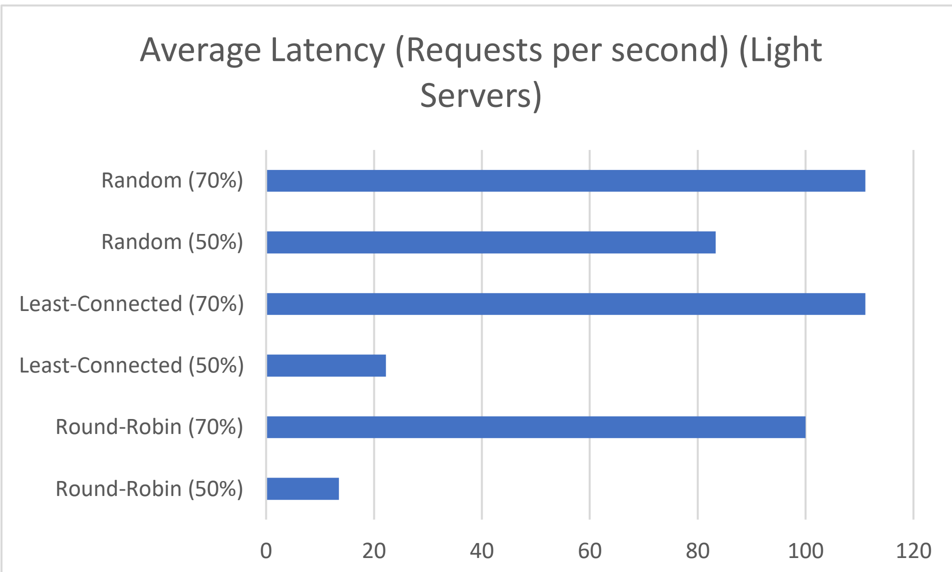
In this script, we're using xargs to send requests (1000 requests were sent for the light server, 100 for the medium, and 10 for the heavy servers) in parallel with the -P10 option specifying the number of processes to run at the same time. We're also using seq to generate a list of numbers from 1 to 10 that will be passed to xargs.

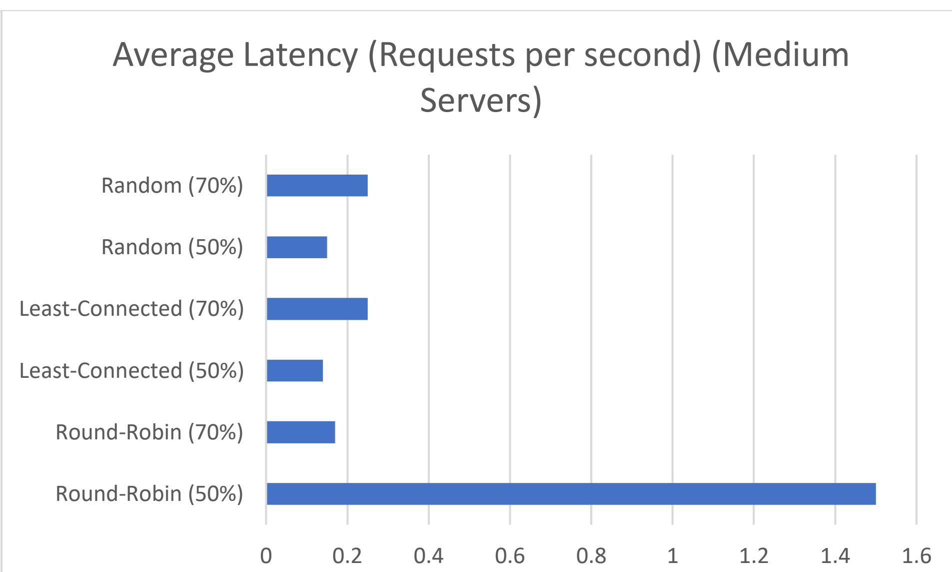
We're then using sh -c to execute a curl command for each number passed by xargs. If the curl command succeeds, it will output the string "success". We're then counting the number of successful requests using wc -l.

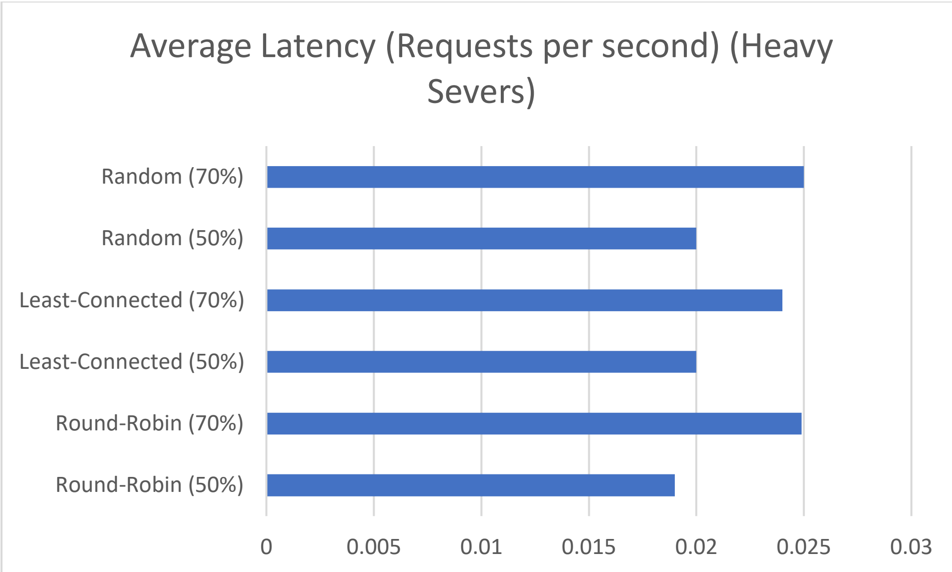
Finally, we're calculating the duration of the requests and outputting the results along with the total count of requests and the number of successful requests. Which is then used to find the Average Throughput and Average Latency per request.

Results

Charts







Data

Server Type	Algorithm	Number of Online Servers	Average Throughput	Average Latency
Light	Round-Robin	50%	0.074	13.51
Light	Round-Robin	70%	0.01	100
Medium	Round-Robin	50%	6.3	1.5
Medium	Round-Robin	70%	5.7	0.17
Heavy	Round-Robin	50%	50.1	0.019
Heavy	Round-Robin	70%	40.5	0.0249
Light	Least-Connected	50%	0.045	22.22
Light	Least-Connected	70%	0.009	111.11

Medium	Least-Connected	50%	6.7	0.14
Medium	Least-Connected	70%	4	0.25
Heavy	Least-Connected	50%	50.1	0.02
Heavy	Least-Connected	70%	40.5	0.024
Light	Random	50%	0.012	83.33
Light	Random	70%	0.009	111.11
Medium	Random	50%	6.7	0.15
Medium	Random	70%	4	0.25
Heavy	Random	50%	50	0.02
Heavy	Random	70%	40.5	0.025

Interpretation

Looking at the data, we can see that for light and medium server types, the Least-Connected algorithm has the lowest average latency and highest throughput across all usage percentages. For heavy server types, the Round-Robin algorithm has the highest throughput at 70% usage but has a significantly higher latency than the Least-Connected algorithm.

As the servers available increases from 50% to 70%, there is a slight decrease in throughput for all server types and algorithms, except for the Heavy server type with the Round-Robin algorithm, which has an increased throughput.

The Least-Connected algorithm performs better in terms of average latency and throughput compared to the Round-Robin and Random algorithms, especially for light and medium server types.

Based on this, it can be concluded that the Least-Connected algorithm is the most suitable algorithm for load balancing, as it provides the lowest latency and highest throughput for all server types. However, for heavy server types, Round-Robin may provide better throughput at the cost of higher latency.