Homework 3

CS 499: Datacenter Computing

Department of Computer Science

University of Cyprus

Homework: 3

Team 3: Ioannis Constantinou (913177)

Georgia Antoniou (918661)

Single Node

Figure 1 presents the throughput and latency of a monolithic application and a microservice application running on a single node. We observe how the latency and achieved throughput changes while increasing the requested throughput from 2000 to 8000 requests per second.

As expected, the latency of the microservice implementation is higher than the latency of the monolithic implementation. This is because the microservice implementation, and specifically grpc, introduces additional overhead to the software stack like for example the client and server stub (machine independent data representation). In this case we do not consider the network overhead of a grpc call, as we evaluate on a single node.

Achieved throughput remains the same for both implementations. This is unexpected, considering that the end to end average response time of the monolithic version is less than the end to end average response time of the microservice. Both setups seem to have dropped queries at 4K and 8K throughput. Our hypothesis is that even though microservice is slower it offers higher parallelism(multiple queries at the same time) and as a result it achieves the same throughput as monolithic configuration.

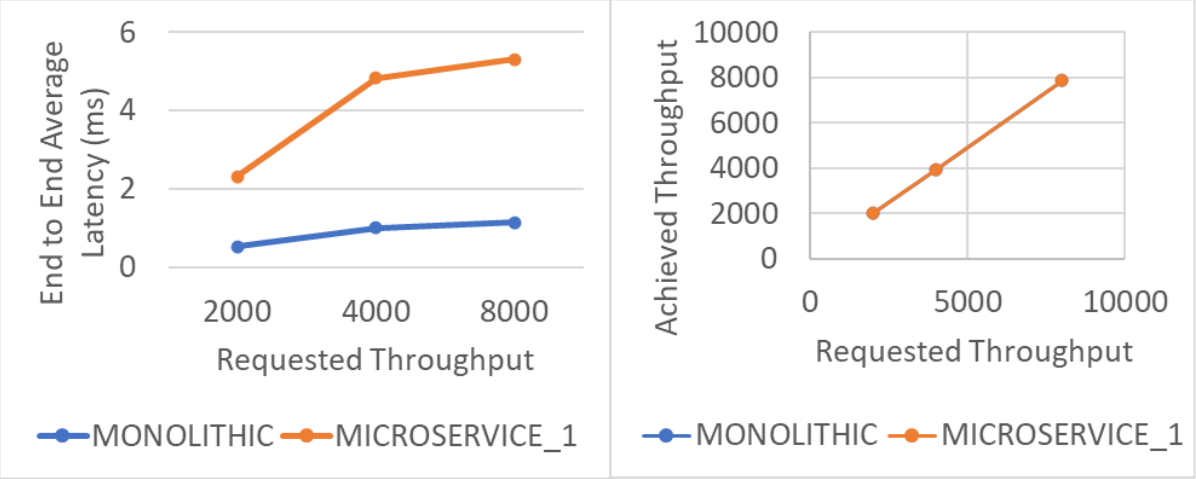

Figure 1: Average Latency and Throughput vs Requested Throughput for single node monolithic and microservice implementation of the Deathstar hotelapp benchmark.

Single Node vs Multi-Node

Figure 2 presents the throughput and latency of a monolithic application and a microservice application running on a single node as well as a microservice running on multiple nodes. We observe how the latency and achieved throughput changes while increasing the requested throughput from 2000 to 8000 requests per second for the hotelapp benchmark of the DeathStar microservice suite.

As expected, the latency of the microservice multi-node configuration is the highest, as it uses grpc, and communicates through the network which increases the end to end average latency. The multi-node configuration achieves the highest throughput, through service replication on a multi-node deployment.

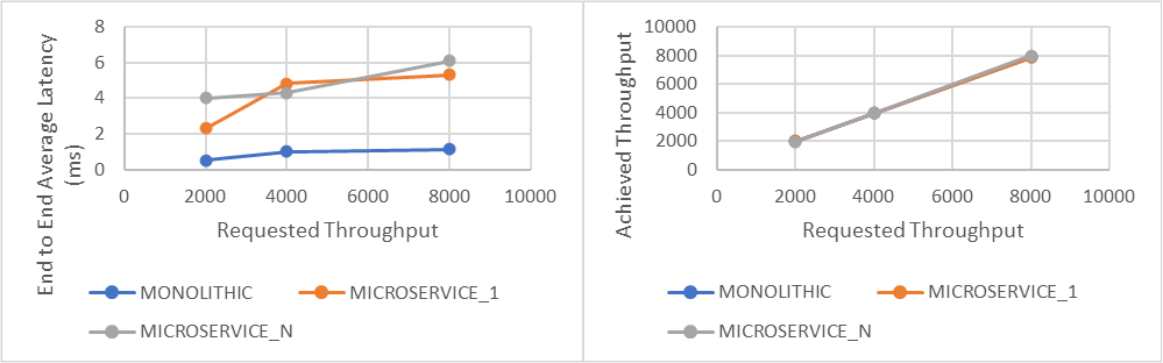

Figure 2: Average Latency and Throughput vs Requested Throughput for single node monolithic and microservice implementation and multinode microservice implementation of the Deathstar hotelapp benchmark.