Evaluating Viability of Network Functions on Lambda Architecture

By Arjun Singhvi, Anshul Purohit and Shruthi Racha

Introduction

The purpose of this document is to provide details regarding the additional experiments that were not presented during the final presentation. As a part of this project, we focused on evaluating the viability of standalone network functions (NFs) on Lambda architectures and proposed a locality-aware, event-based NF chaining system termed as LENS. Apart from the results discussed during the final presentation, the following additional experiments were carried out -

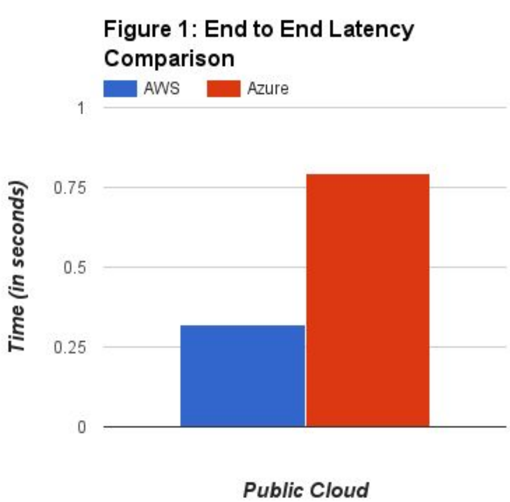
1. Standalone NAT on AWS Lambda vs Standalone NAT on Azure Functions
2. Measurement of Lambda Start-up time on AWS Lambda
3. Investigation of the Lambda instances count on AWS Lambda

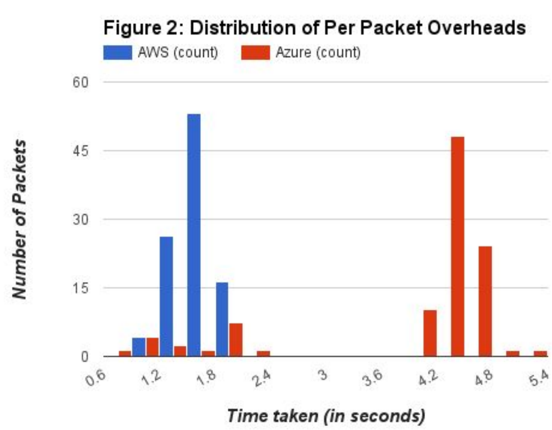
All of the experiments carried out as a part of this project used one of the following synthetic benchmarks -

- Sequential Packet Benchmarker (SPB) - This benchmarker generated a sequence of packets one after the other. The purpose of this benchmarker was to investigate the various factors that contribute to the latency observed to process a packet.
- Concurrent Packet Benchmarker (CPB) - This benchmarker generated a fixed number of packets concurrently. The purpose of this benchmarker was to investigate the scalability offered by Lambda architectures.

We now present the results of the aforementioned experiments.

Standalone NAT on AWS Lambda vs Standalone NAT on Azure Functions





As a part of this experiment we implemented NAT on AWS Lambda as well as Azure Functions. This experiment was carried out to compare the performance of AWS Lambda and Azure Functions by running the SPB and CPB. As seen in Figure 1., The average end-to-end latency observed using the SPB indicates that Azure Functions have higher overhead as compared AWS Lambda. This can be attributed to the relatively higher time incurred by Azure functions to start up the Lambdas and to access the external store. Figure 2., indicates that the overhead incurred by Azure functions is comparatively higher as compared to AWS Lambda during the CPB. This experiments also indicates that AWS Lambda outperforms Azure Functions for the NAT implementation.

This experiment also aided us to decide which Lambda service should we use for the rest of our project. At the end of this experiment we decided to use AWS Lambda for the rest of our project primarily due to the extensive documentation available.

Measurement of Lambda Start-up time on AWS Lambda

In the context of NFs, the top most priority is to ensure that the packets spend minimal time in the NFs. Therefore, it becomes important for us to quantify the startup overhead of the Lambdas. To do so, we ran the SPB on the standalone NAT and compared the end-to-end latency observed by the 1st packet and the average end-to-end latency.

1st packet end-to-end latency (s)	Average end-to-end latency (s)	Slowdown
2.949	0.322	9.16

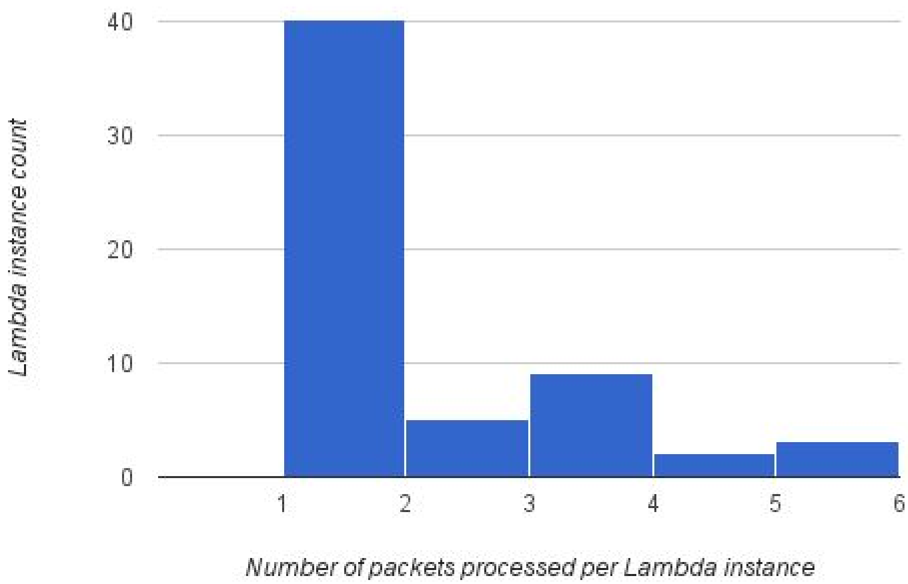
As seen in the above table, there is a non-negligible difference between the latencies. One of the probable reasons for this difference in the latencies is that the startup overhead of the Lambdas dominates for the 1st packet. However, for packets thereon, the warmed-up Lambda instance is used. Therefore, the subsequent packets do not suffer from the high startup overhead. In the context of NFs, it would be necessary to reduce the startup time of these Lambdas to the maximum possible extent to ensure that the packets spend minimal time in the NFs.

Investigation of the Lambda instances count on AWS Lambda

One of the most advertised benefits of Lambda architectures is that it provides infinite scalability. This scalability is possible as the Lambda applications are stateless and the Lambda framework adapts to the changes in the incoming load by launching new Lambda instances when the load is high. As a part of this experiment, we wanted to quantify the number of NAT Lambda instances that AWS Lambda launches during the CPB. The reason for carrying out this experiment was to investigate the variable latency observed by a number of packets during the

CPB. The per-packet latencies indicated by the CPB had a high variability. We suspected that the high latency of a few packets can be attributed to the fact that a new Lambda instance is being launched for each of these packets. On the other hand, the lower latency of the remaining packets is due to the fact that these packets are being processed by warmed-up Lambda instances.

Figure 3: Distribution of Packets Processed per Lambda Instance



As seen from Figure 3., 40 Lambda instances processed only one packet whereas there were a number of other Lambda instances that processed more than one packet during the CPB. The packets processed by warmed-up Lambda instances observe lesser latency. These results explain the reason for the variability in the latency observed.

Conclusion

As a part of the presentation and this write-up, we spoke about the evaluation of the standalone NAT on Lambda architectures. Similar trends were observed across standalone implementations of Firewall and PRADS.