Functions as a Service vs Short-lived containers

Large Systems project proposal

Sean Liao and Mar Badias

November 24, 2019

1 Introduction

Public clouds are growing, and with it comes the latest push into serverless offerings. These come in many forms depending on the abstraction level, but they can largely be grouped into: long-lived containers, short-lived containers, functions as a service.

Functions as a Service (FaaS): Currently the highest level of abstraction, developers provide their application code for the clouds to compile, package, deploy, and run. These are short-lived and stateless, an instance may be started for every request and killed after it completes. Billing is only for the time it is running serving a request. Examples: AWS Lambda, GCP Cloud Functions, Azure Functions, Alibaba Function Compute, IBM Cloud Functions, Zeit Now.

Short-lived containers: These are similar to FaaS: short-lived, stateless runtimes and a similar billing model. Where they differ is that they introduce containers, giving developers control of the execution environment, allowing them to run languages or runtimes unsupported by FaaS. Examples: AWS Fargate, GCP Cloud Run, Azure Container Instance, Alibaba Elastic Container Instance.

Long-lived containers (traditionally Platform as a Service (PaaS)): These can be full-fledged, stateful applications, packaged in containers. The clouds will take these and run them for you on VMs. Auto-scaling and load balancing is usually offered, but fast startup times are not guaranteed. These should be considered an alternative UI to the underlying VMs, which will be reflected in the pricing model (charge for underlying VMs). Examples: AWS Elastic Container service, GCP App Engine, Azure App Services, Alibaba Container Service.

2 Research question

Given the similarities between FaaS and short-lived containers, we want to look into both profiles and answer the following question:

- **When one should be chosen over the other when both are available?**

For deciding which is preferable we will consider the following subquestions:

- **Which has better raw computer performance?** With this question we will determine which solution provides the faster performance. While some platforms do provide numbers, we aim to check if these are comparable across services. Better here would be a faster completion of tasks, and/or a corresponding reduction in costs.
- **Which one has lower platform overhead?** We plan to measure the excess of computation time introduced by the platform as it supposes extra time in client request that we wish to minimize. Lower platform overhead will be considered more preferable.
- **Which one has lower cold start latencies?** When a new instance receives its first request, the response time increases because this instance must be created. As answering the client request needs to be done as fast as possible, lower cold start will be considered preferable.

We want to test the products offered by the top 5 cloud providers as of 2019 [**?**]: Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, Alibaba Cloud, and IBM Cloud. Additionally we want to test Zeit Now, a startup in the FaaS space popular for its streamlined experience. We will not be testing long-lived containers as they require a different application architecture for effective utilization and should be compared to raw VMs (AWS EC2, GCP Compute Engine).

3 Methods

We want to use image processing as our test workload, a real world [**?**] use case. Specifically we will be testing image resizing (thumbnail creation). We aim to use the same code for all platforms (excluding API adapters) and our choice of language is Python because is one of the few languages that all platforms support.

Image processing was selected as it represents a common workload that, without hardware accelerators, relies heavily on CPU performance. It also does not require access to external resources, such as databases, which while also a common workload, introduce too much variability.

We will send images as http requests to the various platforms to be resized. The code we deploy on the platforms will be responsible for both resizing and measuring the time it takes to do so. This will form the basis for our calculations of compute performance. We will additionally measure roundtrip time for requests from the client, this, minus the computation time will be used to calculate the platform overhead. The same experiment will be performed for obtaining the cold start latencies but with different timing to ensure that the instance is created when receiving the request.

Our plan is to spread out testing over a week, to even out variability from running at different times. Specifically we want to test hourly over a week for compute and overhead at single and 50 concurrent requests and repeat it 10 times, to cover the different concurrency guarantees of different products, and every 3 hours for cold start times (to allow time for the function to be evicted from local caches, additional testing required).

At the end of the experiments, the billing of each service will be compared too.

4 Estimated Cost

Most cloud providers offer a free trial between $200 and $300, excepting Amazon Web Services. As explained on section **??**, our experiments will be performed hourly over a week and repeating 10 times a single and 50 concurrent requests. This supposes a total of 85680 request to the cloud service.

Containers as a Service

- **AWS Fargate**
  Amazon does not offer a free trial or a free tier for AWS Fragate. Prices are:
  $0.04048 per vCPU per hour.
  $0.004445 per GB of memory per hour.
  We estimate that, using its lowest configuration (0.25vCPU and 0.5GB of memory) processing an image or a chatbot request will take 5 seconds approximately. So, the total bill would be:
  0.04048*0.25/ 3600 * 85680*5 = $1.20
  0.004445*0.5/ 3600*85680*5= $0.03
  So, in total we expect to spend $1.13 which is 1.12€ .

- **GCP Cloud Run**
  Google offers $300 free trial during one year new Google Cloud Platform clients, which includes us.

- **Azure Container Instance**
  Azure also offers $200 of free trials for new users.

- **Alibaba Elastic Container Instance**
  They also offer $300 of free trial for new users.

Serverless / Functions as a Services

- **AWS Lambda**
  Although AWS does not offer a free trial, AWS Lambda has a free tier which includes 1M free requests per month and 400000 GB-seconds of compute time per month. Specifically, using 128MB memory gives us 3,2M free seconds which is enough for the test we plan to perform.

- **GCP Cloud Functions**
  Besides offering a free trial, Cloud Functions has a free tier which includes 2M free requests, 1M seconds of compute and 400000 GB-seconds of resources. We think that is enough for this project.

- **Azure Functions**
  hey offer a monthly free grant of 1 million requests and 400000 GB-s of resource consumption per month per subscription.

- **IBM Cloud Functions**
  They also offer a free tier which includes 5M executions per month, 128MB of memory and 500ms for execution .

- **Alibaba Function Compute** They offer a monthly free quota of 1M requests, 400000GB-s of resource consumption, which is enough for our project.

- **Zeit Now**
  They offer a free plan which includes 20 hours of resource consumption and 10 seconds of execution duration for request.

5 Related work

Cloud FaaS performance has been subject of previous research. An excellent example is [**?**] where multiple serverless providers are continuously been benchmarked. Their points of comparison will be used in this project for measuring and comparing them short-lived containers. Another example of an excellent comparison of Faas providers is [**?**]. Cloud short lived containers have also been a topic of study, comparing them to long live containers or performance test among of different provides. We can find examples of this in [**?**] and [**?**]. Comparison between Faas and short-lived containers has also been studied but from a functionality point of view, oversimplificating it and not taking into account the performance [**?**][**?**][**?**].

Up to our knowledge, the performance differences between Faas and short live containers have not been comprehensively analysed yet, which motivates our research.

References

[1] Larry Dignan. Top cloud providers 2019: AWS, Microsoft Azure, Google Cloud; IBM makes hybrid move; Salesforce dominates SaaS. `https://www.zdnet.com/article/top-cloud-providers-2019-aws-microsoft-azure-google-cloud-ibm-makes-hybrid-move-salesfo`

[2] Amazon Web Services. Square Enix Case Study. `https://aws.amazon.com/solutions/case-studies/square-enix/`.

[3] Bernd Strehl. Serverless Benchmark. `https://serverless-benchmark.com/`.

[4] Maciej Malawski; Kamil Figiela; Adam Gajek; Adam Zima. Benchmarking Heterogeneous Cloud Functions. `https://www.icsr.agh.edu.pl/~malawski/CloudFunctionsHeteroPar17InformalProceedings.pdf`.

[5] Michael Wittig. ECS vs. Fargate: What's the difference? `https://cloudonaut.io/ecs-vs-fargate-whats-the-difference/`, 2019.

[6] Y.C. Tay ; Kumar Gaurav ; Pavan Karkun. A Performance Comparison of Containers and Virtual Machines in Workload Migration Context. `https://ieeexplore.ieee.org/document/7979796`.

[7] Mike Chan. Containers vs. Serverless: Which Should You Use, and When? https://www.thorntech.com/2018/08/containers-vs-serverless/.

[8] Philipp Muns. Serverless (FaaS) vs. Containers - when to pick which? https://serverless.com/blog/serverless-faas-vs-containers/.

[9] Chad Arimura. Functions vs Containers. https://medium.com/oracledevs/containers-vs-functions-51c879216b97.