Lambada: Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure

[Poster]

Ingo Müller
ingo.mueller@inf.ethz.ch

Renato Marroquín
marenato@inf.ethz.ch

Gustavo Alonso
alonso@inf.ethz.ch

Abstract

The massive, instantaneous parallelism of serverless functions has created a lot of excitement for interactive batch applications. However, due to fundamental limitations of current offerings, there was no consensus yet as to whether or not this architecture is suitable for data processing. We present *Lambada*, a data analytics framework designed for serverless functions, that overcomes the current limitations and simultaneously achieves better performance and lower cost than commercial alternatives.

A full paper on *Lambada* has been accepted to SIGMOD [5].

Poster Summary

Serverless functions have received a tremendous amount of attention since their recent inception. These services allow users to get an extremely high degree of parallelism (in the order of several thousand) within a very short amount of time (typically a few seconds) and are billed at sub-second granularity. Serverless functions thus have the potential to make massive parallelization available to the masses [2]. However, the current offerings have fundamental limitations, leading some people as far as to say that "[they are] particularly bad for data systems innovation" [1].

Through a simple cost model (Block 1 of the poster), we first show what the potential benefit of serverless functions is: While being up to an order of magnitude more expensive than VMs for the example workload of scanning 1 TB from cloud storage, the elasticity of functions allows to start enough of them such that the workload executes in a few seconds, whereas the long start-up time of VMs already takes a large factor longer. Functions are thus only attractive for interactive use cases.

We build a data analytics system, *Lambada* [4], to study the details of the serverless architecture in this context and to quantify the speed/cost trade-off of such a system. In contrast to previous efforts, we design Lambada such that it consists of *only* serverless components (see Block 2). In particular, no billable or user-maintained infrastructure is running *between* queries; the only costs occur for *executing* queries.

The poster shows how Lambada overcomes two of the fundamental challenges inherent in the current cloud function offerings. The first is shown in Block 3. The mere process of *invoking* a number of functions as "workers" that is in the thousands takes in the order of several dozen seconds if done sequentially, thus severely limiting interactivity. The plot shows an example run of the two-level invocation process we designed to overcome this bottleneck. By off-loading the bulk of the invocations to the first level of serverless workers, Lambada can start many thousand workers within a few seconds.

The result is that terabytes of data can indeed be scanned within seconds (see Block 4): Lambada executes the scan-heavy Query 1 of the wide-spread TPC-H benchmark, depending on the configuration, in little more than 10 s and with a cost of around 20 ¢. Both commercial Query-as-a-Service systems we compared with take at the same time longer (BigQuery only marginally so, but Athena more than an order of magnitude) and are substantially more expensive (about one and two orders of magnitude, respectively). For this type of query, data analytics on serverless functions may thus outperform commercial systems by large margins.

For more complex queries, another fundamental challenge needs to be solved (Block 5): Different serverless workers cannot communicate through direct network connections, but need to communicate through files on cloud storage. This approach has been tested by several other authors [2, 3, 6], but rejected as "unfeasible" due to the quadratic number of file creation this entails: For 1 k workers, 1 M files would be created within a few seconds, with is orders of magnitude more than cloud storage systems can sustain.

We show that this is not a fundamental limitation (Block 6). We design a novel shuffle algorithm that shuffles the data in two phases, in each of which the P workers shuffle only within a subgroup of $\sqrt{P}$. This reduces the number of created files to a sub-quadratic amount and allows us to run a complete shuffle among 1 k workers. This is a larger scale than any previous work has achieved and even competitive with solutions that use VMs—while using only serverless components.

References

[1] J. M. Hellerstein et al. "Serverless Computing: One Step Forward, Two Steps Back." In: *CIDR*. 2019.

[2] E. Jonas et al. "Occupy the Cloud: Distributed Computing for the 99%." In: *SoCC*. 2017. DOI: 10.1145/3127479.3128601.

[3] A. Klimovic et al. "Pocket: Elastic Ephemeral Storage for Serverless Analytics." In: *OSDI*. 2018.

[4] I. Müller et al. "Lambada: Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure." In: *SIGMOD*. 2020. DOI: 10.1145/3318464.3389758.

[5] I. Müller et al. "Serverless Clusters: The Missing Piece for Interactive Batch Applications?" In: *SPMA*. 2020. DOI: 10.3929/ethz-b-000405616.

[6] Q. Pu et al. "Shuffling, Fast and Slow: Scalable Analytics on Serverless Infrastructure." In: *NSDI*. 2019.

EuroSys '20, April 27-30, 2020, Heraklion, Greece
https://doi.org/10.3929/ethz-b-000409244