Stateful Entities: Object-oriented Cloud Applications as Distributed Dataflows

Wouter Zorgdrager Kyriakos Psarakis Marios Fragkoulis Eelco Visser Asterios Katsifodimos
Delft University of Technology

Abstract

Programming stateful cloud applications remains a very painful experience. Instead of focusing on the business logic, programmers spend most of their time dealing with distributed systems considerations with the most important being consistency, load balancing, failure management, recovery, and scalability. At the same time we witness an unprecedented adoption of modern dataflow systems such as Apache Flink [3], Google Dataflow, and Timely Dataflow [6]. These systems are now performant and fault-tolerant, and they offer excellent state management primitives [2, 1, 5].

With this line of work, we aim at investigating the opportunities and limits of compiling general purpose programs into stateful dataflows. Given a set of easy to follow code conventions, programmers can author *stateful entities*, a programming abstraction embedded in Python. We present a compiler pipeline, named StateFlow, to analyze the abstract syntax tree of a Python application and rewrite it into an intermediate representation based on stateful dataflow graphs. StateFlow compiles that intermediate representation to a target execution system: Apache Flink and Beam, AWS Lambda, Flink's Statefun [4], and Cloudburst [7]. Through an experimental evaluation, we demonstrate that the code generated by StateFlow incurs minimal overhead. While developing and deploying our prototype, we came to observe important limitations of current dataflow systems in executing cloud applications at scale.

References

[1] A. Akhter, M. Fragkoulis, and A. Katsifodimos. Stateful functions as a service in action. In *VLDB*, 2019.

[2] P. Carbone, M. Fragkoulis, V. Kalavri, and A. Katsifodimos. Beyond analytics: The evolution of stream processing systems. In *SIGMOD*, 2020.

[3] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas. Apache flink tm : Stream and batch processing in a single engine. In *IEEE Data Eng. Bull.*, 2015.

[4] M. de Heus, K. Psarakis, M. Fragkoulis, and A. Katsifodimos. Distributed transactions on serverless stateful functions. In *DEBS*, 2021.

[5] A. Katsifodimos and M. Fragkoulis. Operational stream processing: Towards scalable and consistent event-driven applications. In *EDBT*, 2019.

[6] D. G. Murray, F. McSherry, R. Isaacs, M. Isard, P. Barham, and M. Abadi. Naiad: a timely dataflow system. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pages 439–455. ACM, 2013.

[7] V. Sreekanti, C. Wu, X. C. Lin, J. Schleier-Smith, J. Gonzalez, J. M. Hellerstein, and A. Tumanov. Cloudburst: Stateful functions-as-a-service. In *VLDB*, 2020.