Benchmarking Scalability of Stream Processing Frameworks Deployed as Microservices in the Cloud (Abstract)

Sören Henning[1] Wilhelm Hasselbring[2]

Abstract: This contribution has been published in the *Journal of Systems and Software* (Elsevier) [HH24], `https://doi.org/10.1016/j.jss.2023.111879`.

Keywords: Stream Processing; Microservices; Benchmarking; Scalability; Performance Engineering

1 Summary

Combining distributed stream processing with microservice architectures is an emerging pattern for building data-intensive software systems [He21]. In such systems, different microservices communicate with each other through asynchronous messages and employ stream processing frameworks to continuously process massive amounts of data in a distributed fashion. While there are several frameworks promoting scalability as a core feature, there is only little empirical research evaluating and comparing their scalability.

Our study addresses this gap in the literature and obtains evidence about the scalability of state-of-the-art stream processing frameworks in different execution environments and regarding different scalability dimensions. For this purpose, we employ benchmarking as empirical standard in software engineering research [Ha21], our previously published Theodolite scalability benchmarking method [HH21a; HH22], and our Theodolite stream processing benchmarks [HH21b]. This way, we benchmark the five modern stream processing frameworks Apache Flink, Apache Kafka Streams, Apache Samza, Hazelcast Jet, and the Apache Beam SDK in a systematic way. In total, we conduct over 740 hours of experiments on Kubernetes clusters in the Google cloud and in a private cloud, where we deploy up to 110 simultaneously running microservice instances, which process up to one million messages per second.

We find that all benchmarked frameworks exhibit approximately linear scalability for most use cases as long as sufficient cloud resources are provisioned. However, the frameworks show considerable differences in the rate at which resources have to be added to cope with increasing load. There is no clear superior framework. Instead, depending on the use case Flink, Hazelcast Jet, or Kafka Streams show the lowest increase in resource demand.

[1] JKU/Dynatrace Co-Innovation Lab, LIT CPS Lab, Johannes Kepler University Linz, Linz, Austria soeren.henning@jku.at

[2] Software Engineering Group, Kiel University, Kiel, Germany hasselbring@email.uni-kiel.de

doi:10.18420/sw2024_40

Despite recently proposed performance optimizations, implementations with the Apache Beam abstraction layer have a significantly steeper increase in resource demand compared to all other frameworks, regardless of the use case and whether they are executed by Samza or Flink. We observe our results irrespective of scaling load on a microservice, scaling the computational work performed inside the microservice, the selected cloud environment, and whether the microservice is scaled over multiple nodes or on a single node. The latter means that vertical scaling distributed stream processing frameworks can—to some extent—also complement horizontal scaling. All observed scalability limits seem to be caused by utilized hardware and not by the stream processing frameworks, which means that limits can be raised by using larger clusters.

We conclude that while scalable microservices can be designed with all evaluated frameworks, the choice of a framework and its deployment has a considerable impact on the cost of operating it.

Data Availability A replication package is available at Zenodo (`https://doi.org/10.5281/zenodo.7497280`). The source code of our benchmarks and the associated tooling is available at GitHub (`https://github.com/cau-se/theodolite`) and the software documentation at `https://www.theodolite.rocks`.

Literatur

[Ha21] Hasselbring, W.: Benchmarking as Empirical Standard in Software Engineering Research. In: International Conference on Evaluation and Assessment in Software Engineering (EASE 2021). ACM, S. 365–372, Juni 2021.

[He21] Henning, S.; Hasselbring, W.; Burmester, H.; Möbius, A.; Wojcieszak, M.: Goals and measures for analyzing power consumption data in manufacturing enterprises. Journal of Data, Information and Management 3/1, S. 65–82, 2021.

[HH21a] Henning, S.; Hasselbring, W.: How to Measure Scalability of Distributed Stream Processing Engines? In: Companion of the ACM/SPEC International Conference on Performance Engineering. ACM, S. 85–88, Apr. 2021.

[HH21b] Henning, S.; Hasselbring, W.: Theodolite: Scalability Benchmarking of Distributed Stream Processing Engines in Microservice Architectures. Big Data Research 25/100209, S. 1–17, Juli 2021.

[HH22] Henning, S.; Hasselbring, W.: A Configurable Method for Benchmarking Scalability of Cloud-Native Applications. Empirical Software Engineering 27/143, S. 1–42, 2022.

[HH24] Henning, S.; Hasselbring, W.: Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud. Journal of Systems and Software 208/, S. 111879, 2024.