Tutorial: Distributed computing with live streaming data

Adrian Kosowski Krzysztof Nowicki Przemysław Uznański
Pathway.com
Paris, France

ABSTRACT

"The only constant thing in life is change". A lot of modern data processing applications work with data streams and changing data inputs, and their objective is to provide up-to-date outcomes with low latency at high data throughput.

In this tutorial, we look how to design dynamic algorithms in a systematic way, and to implement them in an actual distributed streaming system. A major challenge here is the design of dynamic algorithms ready for different input data scenarios: data streams with insertion, deletion, arrival of data out-of-order, backfilling, etc.

We center the discussion around designing iterative graph algorithms for time-changing data. For this task, we provide examples of code in industry-standard frameworks (Apache Flink, Spark Structured Streaming, Kafka Streams), as well as in Pathway. Pathway is a new performant data processing framework, for bounded and unbounded data streams, equipped with a Table API in Python, and powered by a distributed incremental dataflow in Rust. It is particularly well suited for implementing "local-type" algorithms.

In the course of a hands-on code tutorial, you will learn how to make a fully functional streaming application. We will write an unsupervised graph learning algorithm, and do a quick integration of data sources, and presentation of outputs. When the application is deployed in streaming mode, it will take care of updating classification outcomes automatically as new data arrives.

We will close with some remarks on consistency and correctness promises that can be asked of distributed streaming systems when executing dynamic algorithms.