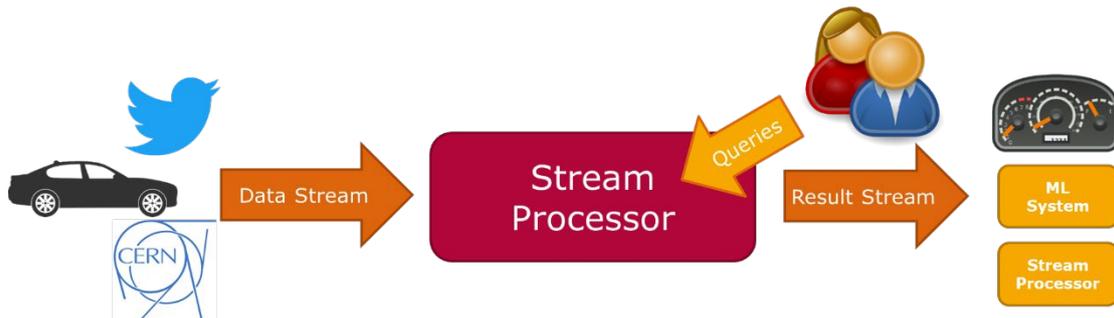


Distributed Dynamic Stream Processing

(Master project, Winter 2020/21)

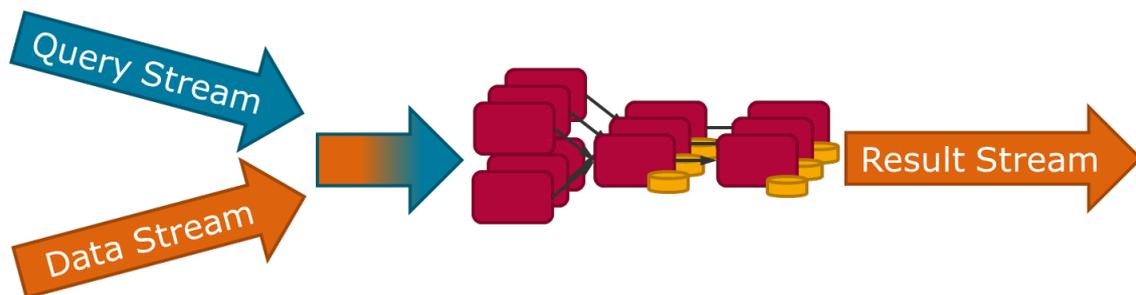
The digital revolution leads to ever increasing amounts of data and a massively increased pace of data generation. In many use cases, archival of the data and later processing is either impossible or uneconomic due to the speed and amount of the data and the quick loss in value of data analysis over time. This has led to the development of stream processing engines (SPE), which can analysis large amounts of data in motion. This leads to two major challenges, the handling of time and potentially endless streams.



Current systems, such as Apache Spark Streaming or Apache Flink, handle these two challenges but work under the strong assumption that an analysis job is running very long and in isolation. This has led to an execution model that is very static regarding queries. Previous work [1] and two previous master projects have explored the option to break this assumption and deal with streams of query additions and removals. Considering a standardized query structure, this leads to orders of magnitude improvements in throughput comparing to state of the art SPEs.

Project Outline

Goal of this master project is to build a prototype of a distributed stream processing engine that has a concept of dynamic query deployment and removal. In this project, a compilation-based approach will be built, with a clear focus on performance.



The prototype should be able to process simple stream processing queries and streams. The set of query operators to be supported will be retrieved from benchmarks such as Nexmark [3] and previous implementations. To speed up the processing, the query dataflows needs to be compiled into binaries. The idea is to generate code for the dataflow [8,9] that can run in a distributed setup. For this, a focus is on distribution and efficient handling of networking. Different techniques can be explored, such as remote direct memory access (RDMA) [5]. In a previous master project, students have explored the compilation-based approach for a single node setup. Students can build on the results of this project.

In this project, students will learn the inner workings of stream processors and data management systems in general, with a particular focus on distribution and query compilation. It is targeting students interested in acquiring skills in data management, stream processing, data flows, compilers, and low-level systems programming.

General information and an introduction on stream processing can be found in the O'Reilly blog posts by Tyler Akidau [5,6] and the stream processing book [7].

Grading

Courses applicable: ITSE (Masterprojekt), DE (Data Engineering Lab)

Graded activity:

- Implementation / group work
- Final report (8 pages, double-column, ACM-art 9pt conference format)
- Final presentation (20 min)

Contact

Tilman Rabl

Literature

[1] Jeyhun Karimov, Tilman Rabl, Volker Markl: AStream: Ad-hoc Shared Stream Processing. SIGMOD 2019. <https://jeyhunkarimov.github.io/assets/publications/karimov-astream-ad-hoc-shared-stream-processing.pdf>

[2] Jeyhun Karimov, Tilman Rabl, Volker Markl: AJoin: Ad-hoc Stream Joins at Scale. VLDB 2020. <https://jeyhunkarimov.github.io/assets/publications/karimov-ajoin-ad-hoc-stream-joins-at-scale.pdf>

[3] Pete Tucker, Kristin Tufte, Vassilis Papadimos, and David Maier: NEXMark—A Benchmark for Queries over Data Streams (DRAFT). Technical report, OGI School of Science & Engineering at OHSU, 2008.

[5] Hendrik Makait: Rethinking Message Brokers on RDMA and NVM. SIGMOD 2020. <https://dl.acm.org/doi/10.1145/3318464.3384403>

[5] Tyler Akidau: Streaming 101. <https://www.oreilly.com/ideas/the-world-beyond-batch-streaming-101>

[6] Tyler Akidau: Streaming 102. <https://www.oreilly.com/ideas/the-world-beyond-batch-streaming-102>

[7] Tyler Akidau, Slava Chernyak, Reuven Lax: Streaming Systems. O'Reilly. <http://streamingsystems.net/>

[8] Thomas Neumann: Efficiently compiling efficient query plans for modern hardware, VLDB 2011: <http://www.vldb.org/pvldb/vol4/p539-neumann.pdf>

[9] Tiark Rumpf: A PL & Compiler View on Data Management and ML Systems. <https://www.tele-task.de/lecture/video/7943/>