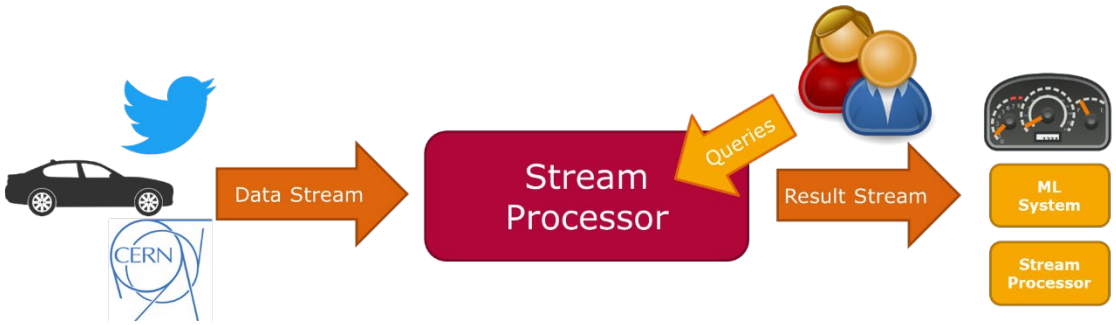
Compilation Techniques for Dynamic Stream Processing

(Master project, Summer 2020)

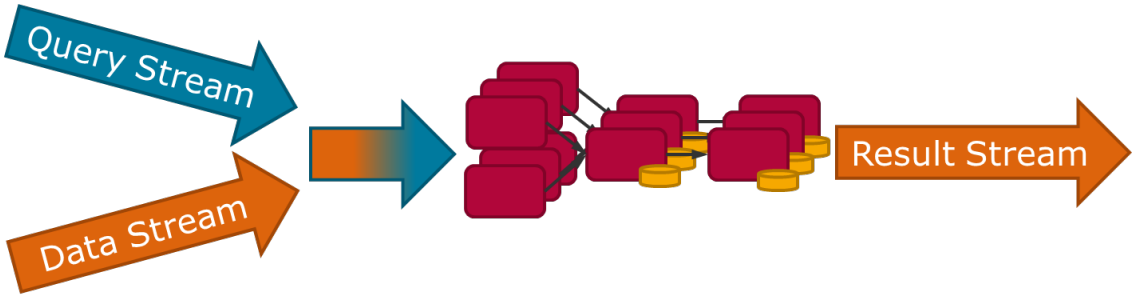
The digital revolution leads to ever increasing amounts of data and a massively increased pace of data generation. In many use cases, archival of the data and later processing is either impossible or uneconomic due to the speed and amount of the data and the quick loss in value of data analysis over 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. Preliminary work [1] and a previous master project has 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 stream processing engine that has a concept of dynamic query deployment and removal. Unlike previous work [1], which is built on top of the SPE Apache Flink [2], and the 2019 master project, 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 [2], LinearRoad [3], or TPCx-IoT [4] and previous implementations. To speed up the processing, the query dataflows need to be compiled as binaries. The idea is to generate code for the dataflow and then try different compilation techniques [8,9]. As an extension, an incremental, dynamic approach combining different compilation techniques can be used. In this setup, a quick, less efficient compilation or

interpretation can incrementally be replaced with more highly optimized, yet more slowly compiled versions of the query plan.

In this project, students will learn the inner workings of stream processors and data management systems in general, with a particular focus on 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

Tilmann Rabl

Literature

[1] Jeyhun Karimov, Tilmann 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] 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.
[3] Arvind Arasu et al.: Linear Road: A Stream Data Management Benchmark - https://www.cs.brandeis.edu/~linearroad/
[4] TPC Express Benchmark IoT (TPCx-IoT) - http://www.tpc.org/tpc_documents_current_versions/pdf/tpcx-iot_v1.0.3.pdf
[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 Rompf: A PL & Compiler View on Data Management and ML Systems. https://www.tele-task.de/lecture/video/7943/