Data stream processing in realtime environments

Opportunities and challenges in using data analytics in time critical applications

Yannnick Bäumer - Masterstudent

Abstract

Applying data analysis and engineering to the domain of realtime applications becomes steadily more attractive. These new appliances demand new constraints to execution environments, programming paradigms and toolchains. Newer developments on the cloud service sector display a trend that could meet these constraints in near future.

Realtime operators in data analysis

With data analysis and machine learning becoming widely used in different domains, also usecases arise which expect strict timing constraints. Weather Analysis and prediction for engine control, industrial controlling to optimize the factory or mining pit throughput [3], controllers for (smart) ernergy grids [3] or image processing for the self-driving car are some examples.
A realtime operator has a priority and a timeframe during which it must be finished. This timeframe (light blue bars in Figure 1) is usually wider than the actual amount of processing time (blue bars in Figure 1), so a realtime scheduler can ensure that all deadlines and dependencies are met.

Data stream architecture

The data stream architecture introduced in [1] yields several properties that help to meet realtime constraints:
It decouples the operator from the database so write- and read-operations are exchanged with a datastream, a simpler datastructure that can better be optimized to timing constraints. The state of the applicatin is kept locally so network delays are omitted. Furthermore dependencies between operators can be modeled (Figure 2) and groups of operators moved to single network segments to reduce the network delay (orange boxes in Figure 2).

Challenges

To implement a realtime operator, timing standards must be met throughout the whole technology stack used for it's implemention (See Figure 3). The underlaying operating system, for example RT-Linux [5] must provide processes with special priorities and a realtime scheduler to meet the timing constraints.
The DSMS (Datastream management system) which replaces the Database Management System must implement similar scheduling algorithms to satisfy dependencies and priorities of the operator.
So that a scheduler can decide, when to schedule an operator, it must posses certain metrics and statistics about the operators runtime behaviour.

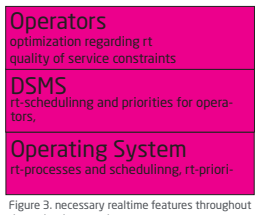

Figure 3. necessary realtime features throughout the technology stack

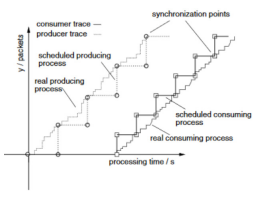

Figure 4. timing behaviour between two connected applications [4]

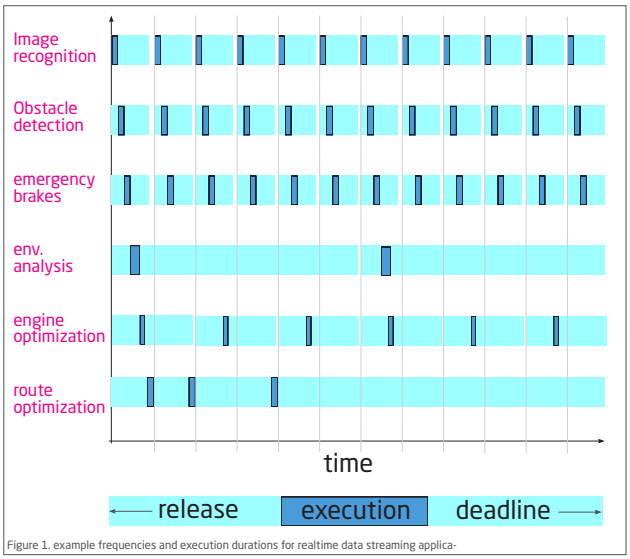

Figure 1. example frequencies and execution durations for realtime data streaming applica-

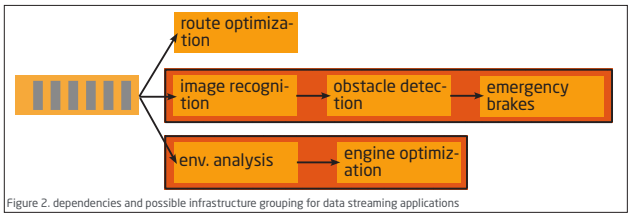

Figure 2. dependencies and possible infrastructure grouping for data streaming applications

[4] discusses different scheduling strategies based on the quality of service properties: execution frequency, mean execution time, worst case execution time and the execution time jitter of the operator.
The paper describes how to schedule operators so that producer and consumer of datastreams are executed in the right order without too much waiting time inbetween (See Figure 4). As the execution time is fixed within the borders of mean execution time and jitter, well tested algorithms like RMS can be used to create optimal schedules [4].

Realization

Realtime applications using operator, as described above, have a higher affinity to work locally and on smaller datasets. The progress that cloud services such as Amazon Redshift Spectrum show concerning the control of the underlying system and it's locality, data storage formats, and optimizing quality of service features [2] will make a growing number of data applicable for realtime applications. The cloud providers could then start to offer services specifically tuned to realtime applications. These services could, for example, be run on the same network segment in a single datacenter closest to the executing realtime application to reduce the network delay.
Furthermore, optimizing an operator to meet the timing constraints mentioned above is not a trivial task. Tools like query languages or operator frameworks would make realtime services available to a wider spectrum of data engineers.

Referenzen

[1] Fabian Hüske, Modern Stream Processing with Apache Flink
[2] Martin Grund, Amazon Redshift Spectrum
[3] Vincent Aït, Ammar, Tobias Wieschnowsky, Data Mining & Predictive Maintenance for Energy efficient coal mining
[4] Sven Schmidt et. al. Real-time Scheduling for Data Stream Management Systems
[5] RT Linux, https://rt.wiki.kernel.org/

Projektbeteiligte

IT-Systems Engineering | Universität Potsdam
Dr. Ralf Krestel , Prof. Dr. Emmanuel Müller, Dr. Matthias Uflacker, Prof. Dr. Felix Naumann

Prof.-Dr.-Helmert-Str. 2-3 I D-14482 Potsdam

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam