Streaming
The New Big Data Rock Star

What is Stream Processing?

Stream processing is a programming paradigm which helps to fast ingest data from continusly producing data sources, like IOT devices or click events from a web site.
Among the most used stream processing frameworks are Apache Spark Streaming, Apache Flink and Kafka Stream. [5] Multiple applications can access and handle a stream of information rather than pulling information from a data warehouse. You can filter information, aggregate it in an application A and process the outcome in another application B. [3,5]

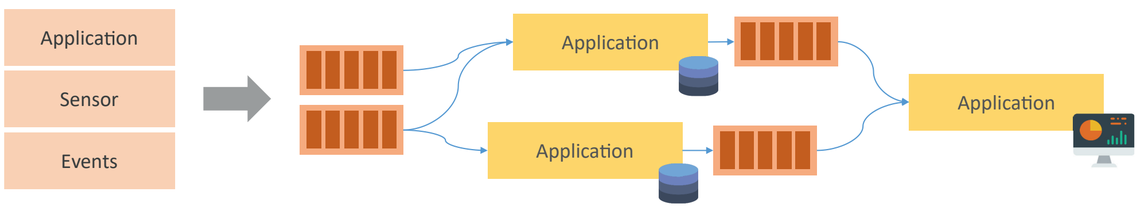

Figure 1: Flow of stream data to multiple applications transforming the data and output a new data stream to an analytics tool.

Why use Stream?

Stream processing is used in most applications because of the performance. The data is analysed before it actually is written down to disk. The latency of batch processing is in the range of minutes to hours whereas the latency of stream processing ranges from seconds to milliseconds. [1]

Is Streaming the Future?

Stream processing can scale better with more nodes and cores. This will have a big impact on future systems as they will have more rather than faster cores. Also, parallelism will enable replaying history data through the stream pipeline in a short time with no need for batch processing. [4,5]

What is Batch Processing?

Batch processing is a programming paradigm where a set of data is collected over time and then fed batchwise into another system. Analysis is then based on the collected data of one batch. This introduces a time lag and real time applications such as analysis application, fraud detection systems cannot work on the latest data. [3]

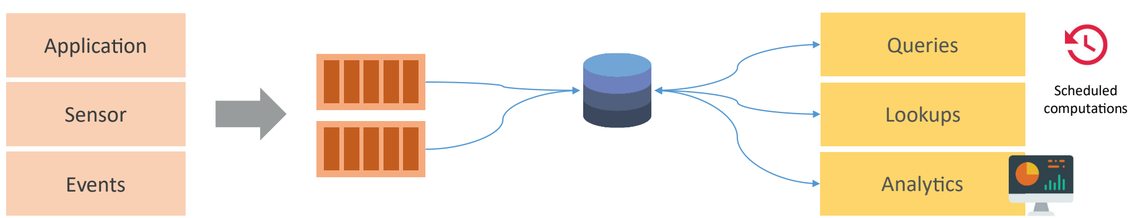

Figure 2: Data is collected in batches and written to a database. Applications then work with the database. The batches are transformed in scheduled computations introducing a time lag.

When Use What?

If your data changes slowly while your queries change often like in analysis and exploration scenarios, batch processing will fit the best. If you have fast changing data and slowly changing queries like in business logic applications, stream processing will be a good choice [3].

Best of Both Worlds?

The lamda architecture combines real time analytics with the possibility to persist the raw input data or integrate an existing data warehouse by forking the stream. One stream is used for stream processing while the other one is saved to the database. This enables an agile and robust data engineering process in which all data can be replayed if an error occurs. However, maintaining two systems and code bases can be challenging and resource consuming. [4]

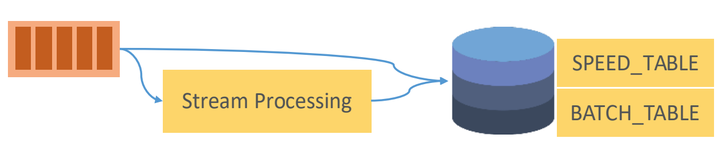

Figure 3: Lambda Architecture

Michael Janke

IT-Systems-Engineering Master
Ringvorlesung Data Engineering in der Praxis

Hasso Plattner Institute, Potsdam, Germany

E-Mail: michael.janke@student.hpi.de

References

[1] as presented in Eating News from the Web
Peter Adolphs, Neofonie
[2] as presented in Queue Mining - Analysis of Clinical Pathways
Matthias Weidlich, Humboldt Universität zu Berlin
[3] as presented in Modern stream processing with Apache Flink
Fabian Hueske, data Artisans
[4] Questioning the Lambda Architecture **Jay Kreps**
https://www.oreilly.com/ideas/questioning-the-lambda-architecture
[5] Distributed Stream systems **Matthias Niehoff**
https://blog.codecentric.de/2017/03/verteilte-stream-processing

Hasso Plattner Institut
Digital Engineering · Universität Potsdam