Stream Processing on Big Data

Stream Processing is an approach to process a continuous and fast incoming stream of data efficiently. It enables applications to execute analytics and to process and filter data on the fly as it comes in. It is most effective for applications where there is a high ratio of event throughput versus changes of queries. Therefore, it is very useful for example in the real-time analytics field where the application can calculate mathematical or statistical analytics on the fly within the stream. Additionally, the typical stream processing architecture allows to easily incorporate external databases and other processing engines.

### Batch Processing

**Use Cases**: ad-hoc queries, data exploration, static data, ML training, (hyper) parameter tuning

- Data changes slowly
- Queries change frequently

### Stream Processing

**Use Cases**: data pipelines, standing queries, anomaly detection, ML evaluation, continuous applications

- Long-lived queries
- Rapid data input

### Stream Processing Architecture

![Stream Processing Architecture Diagram]

### The Future of Stream Processing

Stream processing started off as a means to enable development of real-time analytics platforms with dashboards. Right now however, it is on the verge of advancing in many more territories. As the number of fields for applications of Big Data is increasing so do new use cases for stream processing come up. Especially the combination with deep learning has a lot of potential. Moreover, the most popular Apache frameworks for stream processing (e.g., Spark, Storm, Flink) are rather young. They only got released a few years ago and still leave room for adapting many more use cases.

### Common Use Cases

- Monitoring systems
- Filtering & analyzing news data
- Intelligence and surveillance
- Fraud detection
- Smart order routing
- Risk management

---

Stephan Detje  
Master IT-Systems Engineering  
Data Engineering in der Praxis WS 17/18

E-Mail: stephan.detje@student.hpi.de