Stream Processing

An Overview of the Apache Software Foundation Technologies

Apache Kafka

Apache Kafka is an open source software that enables the storage and processing of data streams via a distributed streaming platform. It provides interfaces for writing data to Kafka clusters, reading data or importing and exporting to and from third-party systems.
The core component of the system is a distributed commit/transaction log. The architecture is a publish-subscribe architecture and consists of a cluster where brokers save messages with a time stamp, called topics. The stored information is replicated and distributed in the cluster.

Apache Flume

Apache Flume is an open source software that enables comprehensive data collection, particularly in the area of streaming event data. The data can be transferred to Hadoop clusters and efficiently processed by that. Flume is mostly used when different data storage devices have to be integrated. It has a flexible and simple architecture based on streaming data flows. A so-called source collects data from a data generator (e.g. Twitter) in the form of events and delivers it to a channel. The channel buffers the data until it a so-called sink collects the data and delivers it to a data storage like HDFS or forward the events to another Flume agent.

Kafka Streams

Kafka Streams is a lightweight, open source library for native event stream processing which is tightly integrated into the Kafka ecosystem. It transforms input Kafka topics into output Kafka topics (or external services). It allows the development of stateful and fully fault-tolerant applications by using an internal changelog topic and a persistent key value store (RocksDB) which saves state to disk and permits for a state larger than main memory. There is no need for time consuming setups or distributed clusters to work with Kafka Streams - if you use Kafka, you can use Kafka Streams.
Although the library is rather new, it is already used by some big companies like Zalando, The New York Times, Pinterest and Trivago for their real time event processing.

Apache Storm

Apache Storm is an open-source, distributed real-time data processing framework.
While being rather simple to use and good for simple event-based use cases, it is extremely fast and can process millions of records per second. Storms fault tolerance makes it highly resilient. It has so called spouts, which represent the source of a data stream in Storm and bolts, which process and transform any number of streams and output new streams. Together spouts and bolts form a so-called topology, which is being set up to create stream processing applications.
Storm was designed to be used with any programming language, which makes it easy to operate for many. It can be deployed as a cluster without huge configuration, but still usable for production use cases. It's being used by Twitter, Groupon and Yahoo for their real-time processing.

Apache Spark

Starting as a university research project in 2009 Apache Spark quickly became one of the most used distributed data processing frameworks. Built as an easier to use and more efficient alternative to Google's MapReduce, it was not cutting back on scalability, fault-tolerance and or distributed processing. Apache Spark was designed with speed in mind and runs lightweight tasks in parallel fashion on a multitude of different cluster configurations like Yarn, Mesos or Kubernetes. Unlike Flink, Kafka Streams or Storm it doesn't support native streaming out of the box, but uses an approach called micro-batching, where incoming tasks are grouped into small batches to increase performance with help of the more traditional approach of batch processing, while still keeping the latency down for each task completion.
Spark is used widely throughout the industry, e.g. Netflix, Yahoo, and eBay have deployed Spark at massive scale.

Apache Flink

Apache Flink, also starting as a university research project, is a highly flexible, open source stream processing framework. It is purely built for native streaming and can run real-time data processing pipelines in a fault-tolerant way at massive scale, while running with minimal latency. Flink uses sophisticated checkpointing and recovery mechanisms to guarantee consistency and durability in its state management which is a key factor when developing a complex stream processing application. It is highly scalable to thousands of cores and runs on Yarn, Mesos or Kubernetes. While being more powerful than Storm and Kafka Streams, it is also the more extensive and sophisticated solution. Flink powers many global enterprises like Alibaba, Comcast, eBay or Uber.

Apache Cassandra

Apache Cassandra is a simple, distributed database management system for very large structured databases. It promises high scalability and reliability in large, distributed systems. The data are stored in key-value relationships. Its masterless architecture enables fast writing and reading at every node. Therefore, Cassandra is well suited for writing and reading several smaller data sets.

Hadoop Distributed File System (HDFS)

The Hadoop Distributed File System is a distributed file system. HDFS is highly fault-tolerant and is designed to be used on low-cost hardware. HDFS offers high throughput access to application data and is suitable for writing large files. It uses a master / slave architecture, with each cluster consisting of a single NameNode that controls the operations for the file system and supports datanodes that manage data storage on individual compute nodes.

Abstract

Stream processing is often used in big data environments or in digitized processes in Industry 4.0.
Instead of first collecting data, storing it in databases and only subsequently processing or analyzing it, as required, stream processing enables data to be processed and analyzed almost in real time. Compared to conventional batch processing, stream processing offers several advantages such as less storage space, real-time reaction options or no interruption of data streams.
There are numerous software solutions for stream processing. The poster is divided into three sections following the analogy of real water streams, reservoirs and sinks. In the section with the flow it tries to provide an overview of the various stream processing technologies of the Apache Software Foundation.
In addition, to allow processing with said technologies, the poster displays a few software solutions which store and transport data streams to and from processing engine. The reservoir section contains selected systems for aggregating and collecting large volumes of data. The beach section contains systems for storage of the massive amounts of data. But note that the first and last section is not completely disjoint. Some systems could be in both sections, so the arrows in the flow show in both directions.

Poster for the Lecture Series on Practical Data Engineering WS19/20

Inspired by the lecture
Apache Flink® An Introduction and Outlook into the Future by Arvid Heise

Alina Gries
IT-Systems Engineering (Bachelor)

Instructors:
Prof. Dr. Tilmann Rabl
Prof. Dr. Felix Naumann

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam