# A Survey of Data Stream Processing Tools

- Pooja Shekhar(23)

## Keywords

Distributed Streaming Processing Systems, MapReduce, Batch Processing, Delivery Semantics

## Abstract

We have a new class of web applications coming up like – Monitoring/Surveillance Systems, Netflix etc. where streams of data is generated with every event. Huge amount of data is generated every instant and it becomes very difficult for the conventional systems in place to process the data streams which led to the rise of a new class of systems called “Distributed Stream Processing Systems” which has the potential to exert backpressure when its unable to process the stream with the pace it gets generated or in other words regulate the speed of the stream. We have several paradigms already at work like MapReduce but this remains a batch processing system again. Since the hardware cost has reduced at price, so it’s achievable to construct a distributed system. There are different data models to process these streams/ event based tuples.

## Stream Processing Model

A stream is a sequence of tuples generated at any time $(a_1, a_2 ...... a_n, t)$ where $a_i$‘s are tuples and t depicts tuples generated any time t. There are processing engines to process these tuples and streams called Processing Engine (PE).These PE’s form a DAG of machines while stream flow through it.

There are certain characteristics that mark the qualities of a good distributed systems:-

Fault Tolerant, Scalability, Concurrency etc.

There are many DSPS available in the market these days: - Apache Storm, Apache Samzaa, Apache Heron, Summingbird, Spark, Flink.

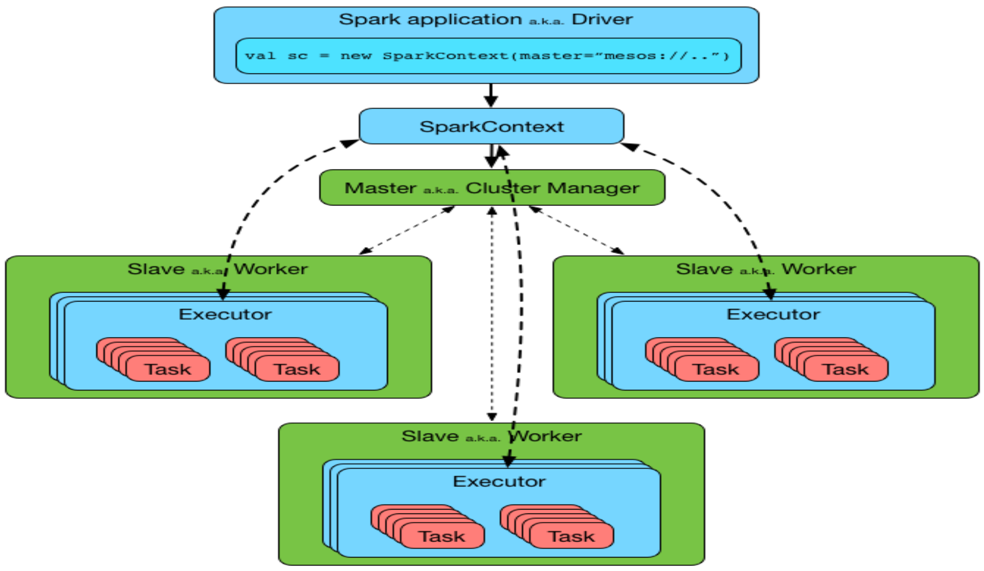
The above systems can be categorized in three broad and overlapping categories – Batching, Micro batching and Real time streaming systems. Spark , Summingbird, Flink and Samzaa overlap the category of (Micro)Batching and Real-time streaming whereas Storm, Heron are purely real time systems.

### Apache Spark

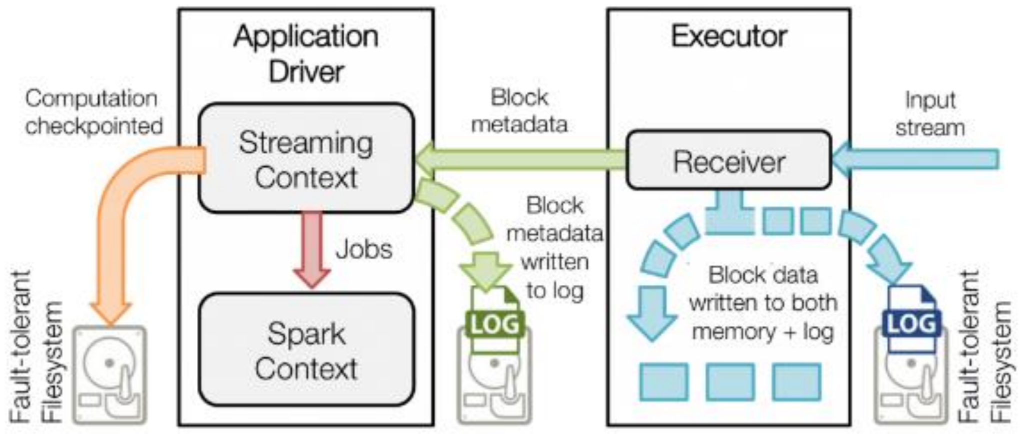
This project was developed at AMPLab at UC Berkley and now it is open sourced by Apache. It’s the most talked about framework that took over the big data world after Storm.

## Architecture

Spark has master/slave architecture. Drivers and Executors run in separate JVM processes.



## Fault Tolerance



([https://databricks.com/blog/2015/01/15/improved-driver-fault-tolerance-and-zero-data-loss-in-spark-streaming.html](https://databricks.com/blog/2015/01/15/improved-driver-fault-tolerance-and-zero-data-loss-in-spark-streaming.html))

RDD abstraction in Spark is lends fault tolerance in Spark. Spark Streaming is built on Spark Context. So mainly, driver program needs to be made fault tolerant. All data operations are periodically check pointed in Spark-in memory to ensure recovery in case of failure using a mechanism called write ahead logs.

Computation Model

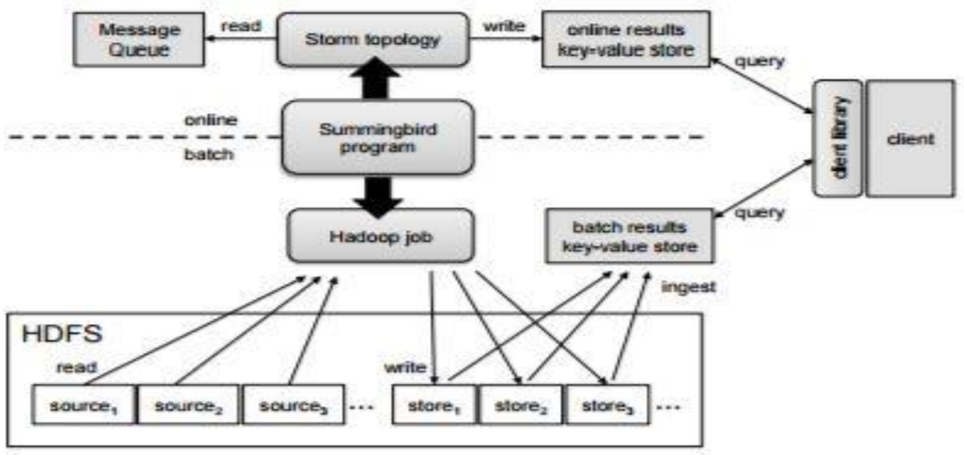
The most important point of focus in Spark is **Resilient Distributed datasets** (RDD) which helps in achieving task parallelism. **Transformations and actions** are another important points of focus that adds to the wisdom of Spark as transformations are not executed until actions need to be performed. This lazy execution adds to the efficiency of the system.

## Twitter’s Summingbird

This system was developed to integrate Hadoop batch processing with processing of online streams. It was developed out of the need when same bit of code needed to be rewritten for batch and online systems. Therefore, engineers at Twitter came up with the system that abstracts the data sources /sink.

Architecture

As mentioned Summingbird integrates batch processing and online streams. Such an architecture is called *Lambda Architecture*.



Fault Tolerance

Since it integrates batch and online processing therefore ,it ensures at least once delivery semantics for online processing and stores the data on disk in batch.

Computational Model

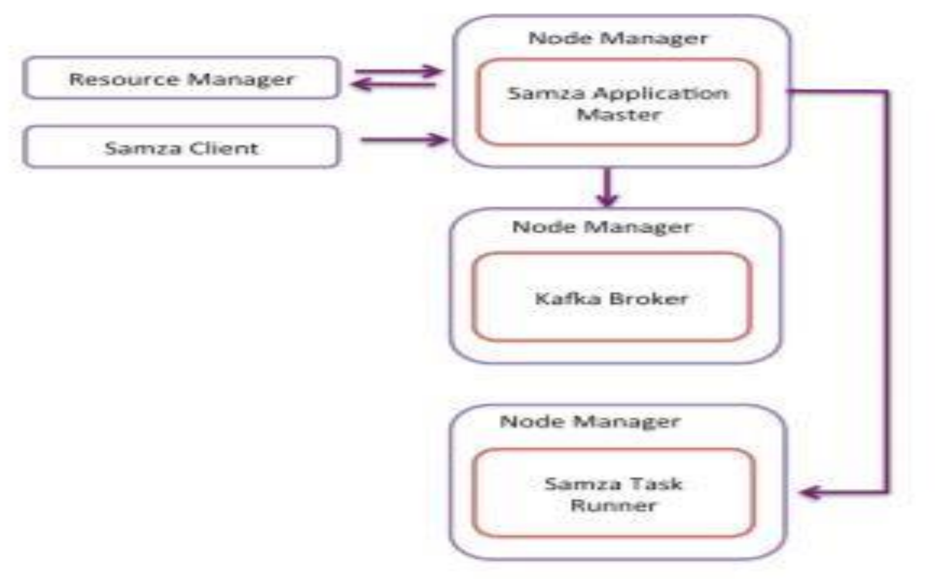
It uses a special algebraic data structure which is at least has a property of *commutative semigroups*. Commutativity is needed because partial results of batch/online computation may be aggregated in arbitrary order.

## Apache Samzaa

Apache Samzaa was developed as a project at Linkedin. Samzaa is made of three layers – streaming layer (Kafka) ,execution layer(YARN) and processing layer(Samza API).It is an immutable message stream produced by Kafka.

### Architecture

Yarn handles resource allocation and Kafka is used for producing streaming messages. Samzaa API is used to run tasks on YARN. Samzaa tasks act as producer and consumer for Kafka. The messages streams are stored in the filesystems of the broker nodes.



### Fault Tolerance

Samzaa provides ‘at least once’ delivery semantics, therefore, it has mechanism to keep the messages in backup in case of failure. But if a broker node fails in Samzaa, it loses the messages in its file system.

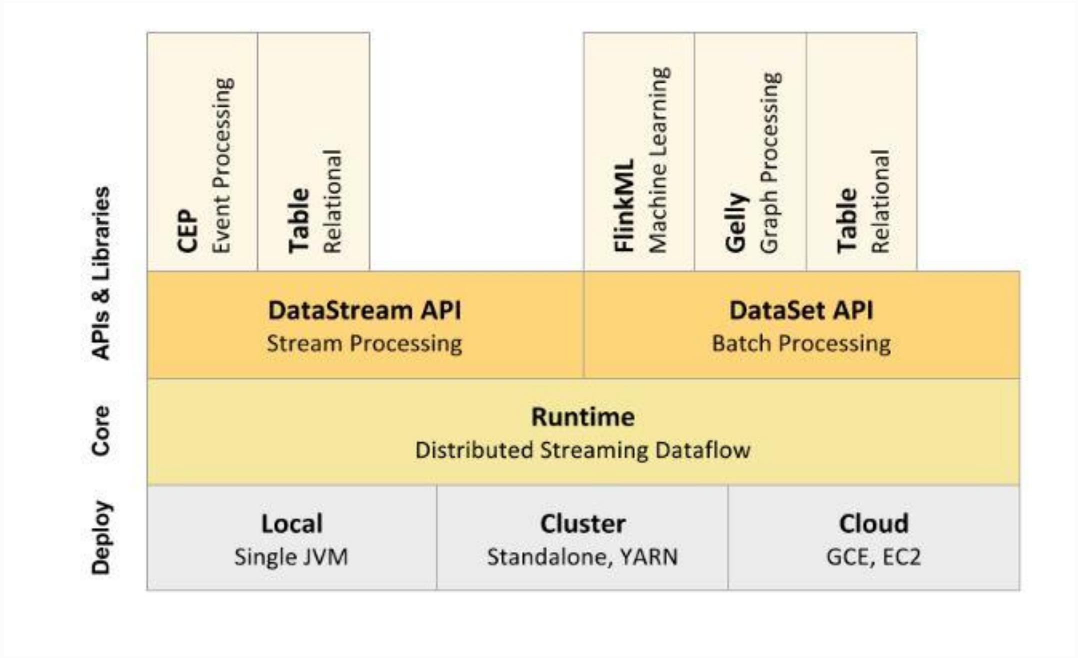
### Computational Model

Samzaa is written in Java. The Samzaa tasks streams are immutable and used for producing and consuming messgaes by Kafka. It’s a pull based model.

## Apache Flink

Flink is a streaming dataflow engine which is used for data distribution , communication and distribued computations.There are three APIs in the system – DataStream API,dataset API and Table API.This system like Summingbird is a platform that provides integration of batch and online processing of data in the same application.

Architecture



Fault Tolerance

Flink provides reliable messaging system by exactly-once delivery semantics. It takes care of failures by checkpointing and partial re-execution. While checkpointing, Flink takes snapshot of the operators.

Computational Model

Data Streams of Flink is the core abstraction for data exchange between the operators. Dataset API as well as DataStream API use this for regulating data flow.