

Apache Flink Tutorial – Learn Flink with Examples

Apache Flink Tutorial

Introduction

In this section of Apache Flink Tutorial, we shall brief on Apache Flink Introduction : an idea of what Flink is, how is it different from Hadoop and [Spark](#), how Flink goes along with concepts of Hadoop and Spark, advantages of Flink over Spark, and what type of use cases it covers.



What is Apache Flink ?

Flink is an open-source framework from Apache Software Foundation designed with the following features:

- Processing distributed (and/or continuous) data (like sensor data, stock market values, etc.)
- High performance
- High Accuracy
- 100% availability
- Low latency in processing (stream-first approach provides large throughput of data)

How is Apache Flink different from Apache Hadoop and Apache Spark ?

- Apache Flink uses Kappa-architecture, the architecture where only streams(of data) are used for processing. Hadoop and Spark uses Lambda architecture, where batches(of data) and micro-batches(of streamed data) are used respectively, for processing.
- Cyclic or iterative processes are optimized in Flink, as Flink has optimization of join algorithms, operator chaining and reusing of partitioning and sorting.

How Apache Flink is related/comparable to Apache Hadoop and Apache Spark ?

Flink is compatible and plays fair along hadoop, distributed datasets, etc., and below are some of the observations:

- Both Flink and Spark are general-purpose platforms for streamed data processing.
- Hadoop and Spark process data in batches. Flink is also able to do batch processing, by only considering batch of data as a stream of data with limits.
- Storm/MapReduce code is compatible to run with Flink execution engine.

- Flink has machine learning module : Flink ML. Spark has machine learning module : Spark MLlib.

Advantages of Flink

Following are the advantages of Apache Flink:

- **Stream-first approach** of Flink provides far better throughput of data, low latency in execution, when compared to micro-batching approach of Spark.
- Flink **handles caching and data partitioning**, whereas in Spark manual optimization is required.
- For data analytics, Flink has **machine learning libraries** (Flink ML), graph processing(Spargel (base) and Gelly(library)), SQL-style querying and **in-memory computation**.

Use-cases for Flink

For reference of use cases that are in live today, refer this link[<https://flink.apache.org/usecases.html>].

Below are some of the use cases from Apache Flink's official website that are in live:

1. E-commerce giant, Alibaba uses Flink to update the product information and inventory info in realtime, to improve the relevancy for its users.
2. Telecom provider, Bouygues Telecom uses Apache Flink to monitor its wired and wireless networks, and therefore enabling a rapid response to outages throughout the country.

Some of the potential use cases are:

1. The usage/statistics of a mobile application could be analyzed in real-time based on geography, climate and such, and making the necessary information available to the users based on the analytics.
2. Flink's stream processing could be used in IOT to process distributed sensory data.

Apache Flink Tutorial

We shall [install](#) Flink and learn its modules.

Conclusion

In this section, we have learnt about Apache Flink, its features, its comparison with Hadoop and Spark, its advantages and finally the use cases.

Apache Big Data Tutorials

↳ [Learn Apache Hadoop](#)

↳ [Learn Apache Spark](#)

↳ [Learn Apache Flink](#)

↳ [Learn Apache Beam](#)

↳ [Learn Apache MXNet](#)

Apache Database Projects

↳ [Apache CouchDB Tutorial](#)

Apache Other Projects

↳ [Apache Tomcat Tutorial](#)

↳ [Apache PDFBox Tutorial](#)

↳ [Learn Apache OpenNLP](#)