CS410 Technology Review

Apache Flink in Recommender Systems

Hongfei Ma (hongfei7)

1. Introduction

A Recommender System refers to a system that is capable of predicting the future preference of a set of items for a user, and recommend the top items. One key reason why we need a recommender system in modern society is that people have too much options to use from due to the huge amount of information overload over internet. To extract useful information, filtering is required. Search engines help to solve this problem to some extent but they do not provide personalization of data. Hence, there is a need of recommender system. With the help of recommender software the preferences of user for a particular product can be foreseen.

In addition, real-time recommendation is also an essential component to a recommender system. For example, Youtube wants to update the recommendation videos for its users as soon as possible based users' watch history. Different people own different histories and we cannot wait for a long time to recommend new videos with batch processing which costs several several hours. Therefore, we want to do real-time calculation and recommendation. Apache Flink is a good choice that helps us to solve this kind of problems.

2. What is Apache Flink

Flink is a distributed processing engine and a scalable data analytics framework. You can use Flink to process data streams at a large scale and to deliver real-time analytical insights about your processed data with your streaming application.

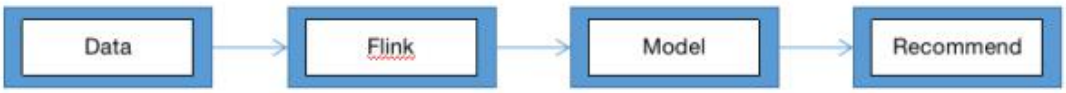
Flink is designed to run in all common cluster environments, perform computations at in-memory speed and at any scale. Furthermore, Flink provides communication, fault tolerance, and data distribution for distributed computations over data streams.

Flink applications process stream of events as unbounded or bounded data sets. Unbounded streams have no defined end and are continuously processed. Bounded streams have an exact start and end, and can be processed as a batch. In terms of time, Flink can process real-time data as it is generated and stored data in storage filesystems. In CSA, Flink is used for unbounded, real-time stream processing.

3. Apache Flink in recommender system

Flink has been used widely around the world. Several large companies use it as a part of their data processing platform. For example, Drivetribe, a digital community founded by

the former hosts of “Top Gear”, uses Flink for metrics and content recommendations. Another example is Amazon Kinesis Data Analytics, a fully managed cloud service for stream processing, uses Apache Flink in part to power its Java application capability.



Here is a basic work flow that shows the procedure of Flink working for recommendation. First of all, huge amount of events create a lot of data. Then, billions of data floods into Flink engine continuously. Flink processes these streaming data and updates its model. Finally, the recommender system push these real-time recommendations to the users. This kind of processes repeat every few minutes so that users can receive the recommendations up to date.

4. Apache Flink vs Apache Spark

Apache Spark and Apache Flink are both open-sourced, distributed processing framework which was built to reduce the latencies of Hadoop Mapreduce in fast data processing. Both Spark and Flink support in-memory processing that gives them distinct advantage of speed over other frameworks.

By the time Flink came along, Apache Spark was already the most popular framework for fast, in-memory big data analytic requirements for a number of organizations around the world. This made Flink appear superfluous. But keep in mind that Apache Flink is closing this gap by the minute. More and more projects are choosing Apache Flink as it becomes a more mature project. Here are some differences between Spark and Flink.

- **Data Processing**: Apache Spark is a part of Hadoop Ecosystem. It is a batch processing system, but it also supports stream processing. While Apache Flink provides a single runtime for both streaming and batch processing.
- **Streaming Style**: Apache Spark Streaming processes data streams in micro-batches. Each batch contains a collection of events that arrived over the batch period. But it is not enough for use cases where we need to process large streams of live data and provide results in real time. In contrast, Apache Flink is the true streaming engine. It uses streams for workloads: streaming, SQL, micro-batch, and batch. Batch is a finite set of streamed data.
- **Performance**: Though Apache Spark has an excellent community background and now It is considered as most matured community. But Its stream processing is not much efficient than Apache Flink as it uses micro-batch processing.Performance of Apache Flink is excellent as compared to any other data processing system.
- **Fault tolerance**: Apache Spark Streaming recovers lost work and with no extra code or configuration, it delivers exactly-once semantics out of the box. Read more about Spark Fault Tolerance. The fault tolerance mechanism followed by Apache Flink is based on Chandy-Lamport distributed snapshots. The mechanism is

lightweight, which results in maintaining high throughput rates and provide strong consistency guarantees at the same time.

- **Computation Model**: Spark has adopted micro-batching. Micro-batches are an essentially ”collect and then process” kind of computational model. Flink has adopted a continuous flow, operator-based streaming model. A continuous flow operator processes data when it arrives, without any delay in collecting the data or processing the data.

5. Conclusion

In conclusion, modern recommender systems require real-time recommendations in some scenarios. Apache Flink is a good choice for these cases because of its streaming-style processing and real-time performance.

References:

https://flink.apache.org/flink-architecture.html
https://docs.cloudera.com/csa/1.2.0/flink-overview/topics/csa-flink-overview.html
https://data-flair.training/blogs/apache-flink-features-why-flink/
https://www.tutorialspoint.com/apache_flink/apache_flink_architecture.htm
https://www.signifytechnology.com/blog/2019/01/is-apache-flink-the-future-of-real-time-streaming-by-anmol-sarna