**CHAPTER 11**

**Where to Go from Here?**

It has been a long journey and you have made it to the end of this book! But your Flink journey has just started, and this chapter points to the possible paths you can take from here. We will provide you with a brief tour of the additional Flink function‐ ality not included in this book and give you some pointers to further Flink resources. There exists a vibrant community around Flink and we encourage you to connect with other users, start contributing, or find out what companies are building with Flink to help inspire your own work.

# The Rest of the Flink Ecosystem

While this book is particularly focused on stream processing, Flink is in fact a general-purpose distributed data processing framework and can be used for other types of data analysis as well. Further, Flink offers domain-specific libraries and APIs for relational queries, complex event processing (CEP), and graph processing.

## The DataSet API for Batch Processing

Flink is a full-fledged batch processor and can be used to implement use cases requir‐ ing one-off or periodic queries on bounded input data. DataSet programs are speci‐ fied as a series of transformations just like DataStream programs with the difference that a DataSet is a bounded data collection. The DataSet API provides operators to perform filtering, mapping, selection, joins, and groupings, as well as connectors to read and write datasets from and to external systems, such as filesystems and data‐ bases. Using the DataSet API you can also define iterative Flink programs that exe‐ cute a loop function for a fixed number of steps or until a convergence criterion is met.

**281**

Batch jobs are internally represented as dataflow programs and run on the same underlying execution runtime as streaming jobs. Currently, the two APIs use separate execution environments and cannot be mixed. However, the Flink community is already working on unifying the two, and providing a single API for analysis of boun‐ ded and unbounded data streams in the same program is a priority in Flink’s future roadmap.

## Table API and SQL for Relational Analysis

Even though the underlying DataStream and DataSet APIs are separate, you can implement unified stream and batch analytics in Flink using its higher-level relational APIs: Table API and SQL.

The Table API is a language-integrated query (LINQ) API for Scala and Java. Queries can be executed for batch or streaming analysis without modification. It offers com‐ mon operators to write relational queries including selection, projection, aggrega‐ tions, and joins and further has IDE support for autocompletion and syntax validation.

Flink SQL follows the ANSI SQL standard and leverages [Apache Calcite](https://calcite.apache.org/) for query parsing and optimization. Flink provides unified syntax and semantics for batch and streaming queries. Due to extensive support for user-defined functions, a wide variety of use cases can be covered by SQL. You can embed SQL queries into regular Flink DataSet and DataStream programs or directly submit SQL queries to a Flink cluster using the SQL CLI client. The CLI client lets you retrieve and visualize query results in the command line, which makes it a great tool to try out and debug Flink SQL queries or run exploratory queries on streaming or batch data. In addition, you can use the CLI client to submit detached queries that directly write their results into external storage systems.

## FlinkCEP for Complex Event Processing and Pattern Matching

FlinkCEP is a high-level API and library for complex event pattern detection. It is implemented on top of the DataStream API and lets you specify patterns you want to detect in your stream. Common CEP use cases include financial applications, fraud detection, monitoring and alerting in complex systems, and detecting network intru‐ sion or suspicious user behavior.

## Gelly for Graph Processing

Gelly is Flink’s graph processing API and library. It builds on top of the DataSet API and Flink’s support for efficient batch iterations. Gelly provides high-level program‐ ming abstractions in both Java and Scala to perform graph transformations, aggrega‐ tions, and iterative processing such as vertex-centric and gather-sum-apply. It also includes a set of common graph algorithms ready to use.

**282 | Chapter 11: Where to Go from Here?**

Flink’s high-level APIs and interfaces are well integrated with each other and with the DataStream and DataSet APIs so that you can easily mix them and switch between libraries and APIs in the same program. For instance, you could extract patterns from a Data‐ Stream using the CEP library and later use SQL to analyze extrac‐ ted patterns or you could use the Table API to filter and project tables into graphs before analyzing them with a graph algorithm from the Gelly library.

# A Welcoming Community

Apache Flink has a growing and welcoming community with contributors and users all around the world. Here are a few resources you can use to ask questions, attend Flink-related events, and learn what people use Flink for:

### Mailing lists

* [*user@flink.apache.org*](mailto:user@flink.apache.org) : user support and questions
* [*dev@flink.apache.org*](mailto:dev@flink.apache.org) : development, release, and community discussions
* [*community@flink.apache.org*](mailto:community@flink.apache.org) : community news and meetups

### Blogs

* [*https://flink.apache.org/blog*](https://flink.apache.org/blog/)
* [*https://www.ververica.com/blog*](https://www.ververica.com/blog)

### Meetups and conferences

* [*https://flink-forward.org*](https://flink-forward.org/)
* [*https://www.meetup.com/topics/apache-flink*](https://www.meetup.com/topics/apache-flink/)

Again, we hope you walk away from this book with a better understanding of the capabilities and possibilities of Apache Flink. We encourage you to become an active part of its community.

**A Welcoming Community | 283**