O'REILLY®

Introduction to Apache Flink

Stream Processing for Real Time and Beyond

Ellen Friedman
& Kostas Tzoumas

O'REILLY®

Introduction to Apache Flink

There's growing interest in learning how to analyze streaming data in large-scale systems such as web traffic, financial transactions, machine logs, industrial sensors, and many others. But analyzing data streams at scale has been difficult to do well—until now. This report delivers a deep introduction to Apache Flink, a highly innovative open source stream processor with a surprising range of capabilities.

Authors Ellen Friedman and Kostas Tzoumas show technical and nontechnical readers alike how Flink is engineered to overcome significant tradeoffs that have limited the effectiveness of other approaches to stream processing. You'll also learn how Flink has the ability to handle both stream and batch data processing with one technology.

- **Learn the consequences of not doing streaming well—in retail and marketing, IoT, telecom, and banking and finance**
- **Explore how to design data architecture to gain the best advantage from stream processing**
- **Get an overview of Flink's capabilities and features, along with examples of how companies use Flink, including in production**
- **Take a technical dive into Flink, and learn how it handles time and stateful computation**
- **Examine how Flink processes both streaming (unbounded) and batch (bounded) data without sacrificing performance**

Ellen Friedman is a solutions consultant and well-known speaker and author, writing mainly about big data topics. She's a committer for the Apache Drill and Apache Mahout projects. Ellen has a PhD in Biochemistry and many years of experience as a research scientist.

Kostas Tzoumas is cofounder and CEO of data Artisans, the company founded by the original creators of Apache Flink. He's got a PhD in Computer Science and has written several technical papers and blog articles on stream processing and other data science topics.

US $24.99 CAN $28.99

ISBN: 978-1-491-97658-6

Introduction to Apache Flink

Stream Processing for Real Time and Beyond

Ellen Friedman and Kostas Tzoumas

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Introduction to Apache Flink
by Ellen Friedman and Kostas Tzoumas

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Shannon Cutt
Production Editor: Holly Bauer Forsyth
Copyeditor: Holly Bauer Forsyth
Proofreader: Octal Publishing, Inc.

Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Panzer

September 2016: First Edition

Revision History for the First Edition

2016-09-01: First Release
2016-10-20: Second Release

978-1-491-97658-6

[LSI]

Table of Contents

Preface

There's a flood of interest in learning how to analyze streaming data in large-scale systems, partly because there are situations in which the time-value of data makes real-time analytics so attractive. But gathering in-the-moment insights made possible by very low-latency applications is just one of the benefits of high-performance stream processing.

In this book, we offer an introduction to Apache Flink, a highly innovative open source stream processor with a surprising range of capabilities that help you take advantage of stream-based approaches. Flink not only enables fault-tolerant, truly real-time analytics, it can also analyze historical data and greatly simplify your data pipeline. Perhaps most surprising is that Flink lets you do streaming analytics as well as batch jobs, both with one technology. Flink's expressivity and robust performance make it easy to develop applications, and Flink's architecture makes those easy to maintain in production.

Not only do we explain what Flink can do, we also describe how people are using it, including in production. Flink has an active and rapidly growing open international community of developers and users. The first Flink-only conference, called Flink Forward, was held in Berlin in October 2015, the second is scheduled for September 2016, and there are Apache Flink meetups around the world, with new use cases being widely reported.

How to Use This Book

This book will be useful for both nontechnical and technical readers. No specialized skills or previous experience with stream processing are necessary to understand the explanations of underlying concepts of Flink's designs and capabilities, although a general familiarity with big data systems is helpful. To be able to use sample code or the tutorials referenced in the book, experience with Java or Scala is needed, but the key concepts underlying these examples are explained clearly in this book even without needing to understand the code itself.

Chapters 1–3 provide a basic explanation of the needs that motivated Flink's development and how it meets them, the advantages of a stream-first architecture, and an overview of Flink design. Chapter 4–Appendix A provide a deeper, technical explanation of Flink's capabilities.

Conventions Used in This Book

This icon indicates a general note.

This icon signifies a tip or suggestion.

This icon indicates a warning or caution.

CHAPTER 1
Why Apache Flink?

Our best understanding comes when our conclusions fit evidence, and that is most effectively done when our analyses fit the way life happens.

Many of the systems we need to understand—cars in motion emitting GPS signals, financial transactions, interchange of signals between cell phone towers and people busy with their smartphones, web traffic, machine logs, measurements from industrial sensors and wearable devices—all proceed as a continuous flow of events. If you have the ability to efficiently analyze streaming data at large scale, you're in a much better position to understand these systems and to do so in a timely manner. In short, streaming data is a better fit for the way we live.

It's natural, therefore, to want to collect data as a stream of events and to process data as a stream, but up until now, that has not been the standard approach. Streaming isn't entirely new, but it has been considered as a specialized and often challenging approach. Instead, enterprise data infrastructure has usually assumed that data is organized as finite sets with beginnings and ends that at some point become complete. It's been done this way largely because this assumption makes it easier to build systems that store and process data, but it is in many ways a forced fit to the way life happens.

So there is an appeal to processing data as streams, but that's been difficult to do well, and the challenges of doing so are even greater now as people have begun to work with data at very large scale across a wide variety of sectors. It's a matter of physics that with

large-scale distributed systems, exact consistency and certain knowledge of the order of events are necessarily limited. But as our methods and technologies evolve, we can strive to make these limitations innocuous in so far as they affect our business and operational goals.

That's where Apache Flink comes in. Built as open source software by an open community, Flink provides stream processing for large-volume data, and it also lets you handle batch analytics, with one technology.

It's been engineered to overcome certain tradeoffs that have limited the effectiveness or ease-of-use of other approaches to processing streaming data.

In this book, we'll investigate potential advantages of working well with data streams so that you can see if a stream-based approach is a good fit for your particular business goals. Some of the sources of streaming data and some of the situations that make this approach useful may surprise you. In addition, the will book help you understand Flink's technology and how it tackles the challenges of stream processing.

In this chapter, we explore what people want to achieve by analyzing streaming data and some of the challenges of doing so at large scale. We also introduce you to Flink and take a first look at how people are using it, including in production.

Consequences of Not Doing Streaming Well

Who needs to work with streaming data? Some of the first examples that come to mind are people working with sensor measurements or financial transactions, and those are certainly situations where stream processing is useful. But there are much more widespread sources of streaming data: clickstream data that reflects user behavior on websites and machine logs for your own data center are two familiar examples. In fact, streaming data sources are essentially ubiquitous—it's just that there has generally been a disconnect between data from continuous events and the consumption of that data in batch-style computation. That's now changing with the development of new technologies to handle large-scale streaming data.

Still, if it has historically been a challenge to work with streaming data at very large scale, why now go to the trouble to do it, and to do

it well? Before we look at what has changed—the new architecture and emerging technologies that support working with streaming data—let's first look at the consequences of *not* doing streaming well.

Retail and Marketing

In the modern retail world, sales are often represented by clicks from a website, and this data may arrive at large scale, continuously but not evenly. Handling it well at scale using older techniques can be difficult. Even building batch systems to handle these dataflows is challenging—the result can be an enormous and complicated workflow. The result can be dropped data, delays, or misaggregated results. How might that play out in business terms?

Imagine that you're reporting sales figures for the past quarter to your CEO. You don't want to have to recant later because you overreported results based on inaccurate figures. If you don't deal with clickstream data well, you may end up with inaccurate counts of website traffic—and that in turn means inaccurate billing for ad placement and performance figures.

Airline passenger services face the similar challenge of handling huge amounts of data from many sources that must be quickly and accurately coordinated. For example, as passengers check in, data must be checked against reservation information, luggage handling and flight status, as well as billing. At this scale, it's not easy to keep up unless you have robust technology to handle streaming data. The recent major service outages with three of the top four airlines can be directly attributed to problems handling real-time data at scale.

Of course many related problems—such as the importance of not double-booking hotel rooms or concert tickets—have traditionally been handled effectively with databases, but often at considerable expense and effort. The costs can begin to skyrocket as the scale of data grows, and database response times are too slow for some situations. Development speed may suffer from lack of flexibility and come to a crawl in large and complex or evolving systems. Basically, it is difficult to react in a way that lets you keep up with life as it happens while maintaining consistency and affordability in large-scale systems.

Fortunately, modern stream processors can often help address these issues in new ways, working well at scale, in a timely manner, and