Apache Flink® and IoT: How Stateful Event-Time Processing Enables Accurate Analytics

Aljoscha Krettek
@aljoscha

dataArtisans

ApacheCon North America
May, 2017
What I’d Like to Talk About

- IoT and event-time stream processing
- Stateful stream processing
- Streaming architecture and Flink
Original creators of Apache Flink®

Providers of the dA Platform, a supported Flink distribution
IoT and Event-time Stream Processing
Example Event Sources
A Simple Definition

IoT use cases from the system’s perspective:

A large number of (distributed) things continuously generating a large amount of data.
IoT: Some Insights

- Data is continuously produced → Stream Processing
- Events have a timestamp → Event-time based processing
- Data/Events can arrive with huge delays/out-of-order
- Most analyses happen on time windows
What Is Event-Time Processing

Event Time

Episode IV 1977
Episode V 1980
Episode VI 1983
Episode I 1999
Episode II 2002
Episode III 2005
Episode VII 2015

Processing Time
What Is Event-Time Processing

![Diagram showing event-time processing with message queue and event timestamps.]

- Event timestamp
- Message Queue
- Processing Time
What’s The Problem?

Mismatch between event time and processing time.
Sources of Time Mismatch

- **Big Mismatch**
  - Network disconnects
  - Slow network

- **Small Mismatch**
  - The nature of distributed systems
  - Differing system clock time
Small Event-Time Mismatch

Robust Stream Processing with Apache Flink®: A Simple Walkthrough
Recap: Event-Time

- IoT use cases need event-time processing
- Even small mismatch of event time/processing time will lead to wrong results
(Stateful) Stream Processing
Stream Processing

Computations on never-ending “streams” of data records (“events”)
Distributed Stream Processing

Computation spread across many machines

Computation

Computation

Computation
Stateful Stream Processing

State is usually partitioned by some key in the data.
Stateful Stream Processing II

- Result depends on history of stream
- A stateful stream processor should give the tools to manage state
  - Recover, roll back, version upgrade, etc.
Recap: Stateful Streams

- Continuous processing of data that is continuously generated
- I.e., pretty much all “big” data
- It’s all about state and time
- Flink does all of that
Operational Issues
Operational Questions

- What happens in case of failures?
- What if I need to update my code/Flink?
- Can I re-process my data?
- How can I execute my programs?
Failure Handling

- JobManager High-Availability using ZooKeeper
- Periodic checkpoints of state to persistent storage (HDFS, S3, …)
- In case of failure: rollback to previous consistent state
Savepoints

- A persistent snapshot of all state
- When starting an application, state can be initialized from a savepoint
- In-between savepoint and restore we can update Flink version or user code
Closing
TL;DR

- Stateful stream processing is nice 😎
- IoT use cases require proper time management
- Apache Flink is a stateful stream processor with plenty of nifty features
Thank you!

@aljoscha
@ApacheFlink
@dataArtisans
Backup Slides
Event-Time Processing
What Is Event-Time Processing

Event Time

Episode IV
1977
Episode V
1980
Episode VI
1983
Episode I
1999
Episode II
2002
Episode III
2005
Episode VII
2015

Processing Time
What Is Event-Time Processing
What is Event-Time Streaming

- Events have timestamps
- Processing depends on timestamps
- An event-time stream processor should give you the tools to reason about time
  - Handle streams that are out of order
IoT use cases need event-time processing

Even small mismatch of event time/processing time will lead to wrong results
History of Flink
A brief History of Flink

- January ‘10: Project Stratosphere (Flink precursor)
- April ‘14: Flink Project Incubation
- December ‘14: Top Level Project
- March ‘16: Release 1.0

Versions:
- v0.5
- v0.6
- v0.7
- v0.8
- v0.9
- v0.10
A brief History of Flink

The academia gap:
Reading/writing papers, teaching, worrying about thesis

January ‘10
Project Stratosphere (Flink precursor)

Realizing this might be interesting to people beyond academia (even more so, actually)

April ‘14
Flink Project Incubation

v0.5 v0.6 v0.7

December ‘14
Top Level Project

v0.8 v0.9 v0.10

March ‘16
Release 1.0

v0.10