7 REASONS TO USE APACHE FLINK

for your IoT Project

How We Built a Real-time Asset Tracking System

Marta Paes Moreira & Jakub Piasecki | ApacheCon Europe | October 2019

FREEPORT METRICS

Marta Paes Moreira

@morsapaes (Twitter)
morsapaes (LinkedIn)

Product Marketing Manager
Ververica
(formerly data Artisans)

Jakub Piasecki

@jakubpiasecki (Twitter)
jakubpiasecki (LinkedIn)

Director of Technology /
Software Architect
Freeport Metrics

Agenda

What is special about IoT data and applications?

Why is Apache Flink a good fit for your IoT use case?

Building an Asset Tracking System with Flink

More IoT use cases for Flink

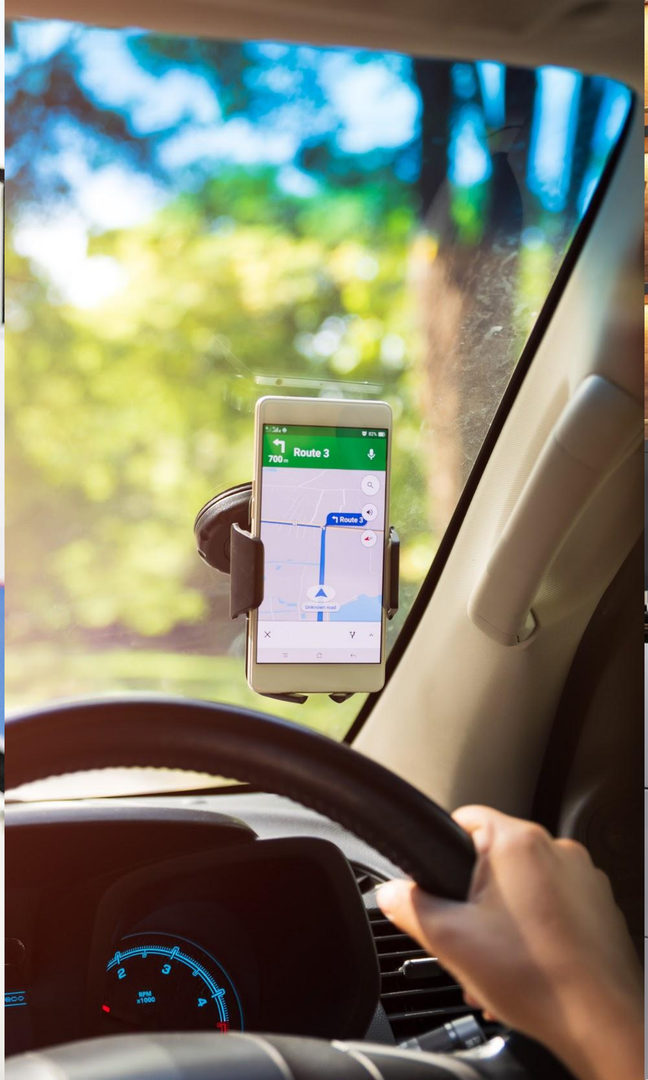
What is so special

about IoT data and applications?

How much IoT data have you generated **this month?**


Running
Stats
kilometers 9.12
duration 49.18
min/km 5.39
calories 498

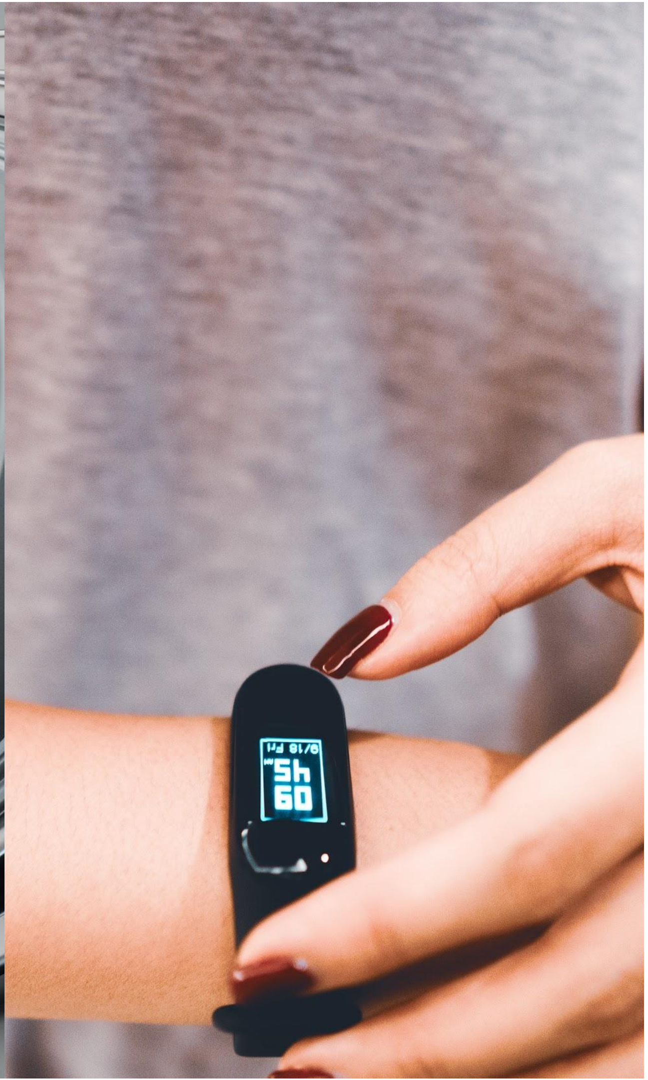
How much IoT data have you generated **this week?**


Route 3
700 m
Route 3
RPM x1000


12:45 pm
04-05-18
Home control
Bedroom 22.5°
Kitchen 23.5°
Bathroom 22.5°
195°
0°

How much IoT data have you generated **today?**


03598
kWh




9:41
FaceTime
Wednesday
7
Calendar
Photos
Camera
Mail
Clock
Maps
Weather
"Weather" has been using your location in the background. Do you want to continue allowing this?
If you always allow access to your location, local weather will be kept up to date at all times. If you only allow access while using the app, it will only be updated after you open the Weather app.
Settings
Continue
Health
Home
Wallet
Settings

IoT Data

Everyday life

Smart home devices, wearables, smart vehicles, mobile phone sensors

Industrial use cases

Measuring solar farm data generation, tracking airport luggage, sensors at production line

More and more IoT use cases, sensors and devices

Predictive maintenance
Smart buildings
Smart cities, utilities
Cashier-less stores
Healthcare wearables
Machine sensors, GPS, RFID tags, beacons, cameras, microphones

IoT Data Properties

Machine generated data can be huge

- 500ZB of data produced in 2019 driven by IOT [Cisco's research] (1ZB = 10^21 B)
  - More data produced than can be stored in data centers
- 1 GPS sensor reporting location every 5s produces ~6 mln events/year

Transmission latencies / out-of-order data

- Mobile network connectivity
- Gateway devices
- Buffering
- Failures

Continuous flow

Applications need to react quickly

User don’t want and sometimes cannot wait

IoT data is gathered in the real world

Applications need to react quickly to trigger reactions in the real world

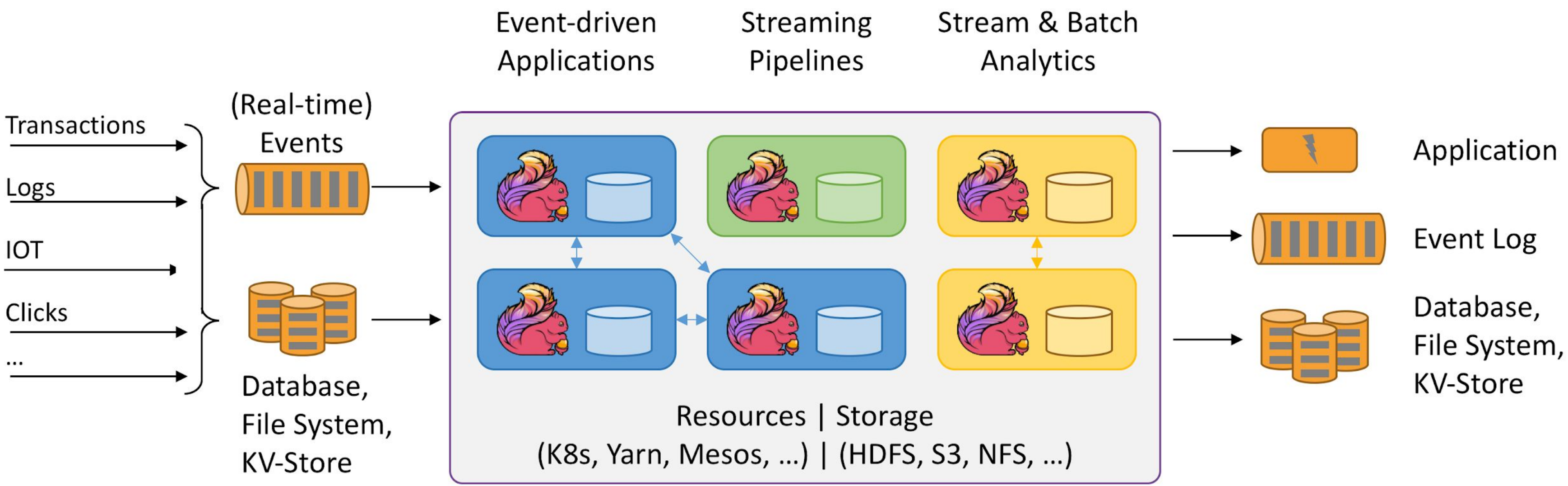
- Send an alert when a valuable asset is leaving a building
- Stop a machine when sensor data indicates an expensive failure in the near future
- Steering city traffic in real time

Apache Flink

The Perfect Fit for IoT Data

What is Apache Flink?

Flink is a distributed system for stateful stream processing. Data can be processed in parallel with low latency.



Reasons to Use It

1

Events are processed with low latency*

*Latency depends on many factors

State is always locally maintained and accessed

- In-memory state backend

Flink’s network stack is optimized for low latency (and high throughput)

- Credit-based flow control mechanism

Support for asynchronous and incremental checkpoints

- Applications can be scaled to many machines to reduce machine load

2

Applications scale to huge data volumes

Streams are partitioned to distribute data and computations

Flink applications can run on 10000+ of cores

- Flink applications can process 5 trillion events per day (~57 million events/ second)

Application state is partitioned (similar to key-value stores)

- Flink application can consistently maintain 20+TB of state
- Application state can be stored on disk (if necessary)

Applications can be scaled in and out

3

Low quality IoT data is handled very well

Delayed or out-of-order data is correctly handled due to event-time processing

- Watermarks control the logical time of an application
- Applications can choose to wait for, redirect, or discard delayed data
- Trade-off result completeness and latency

Inaccurate or imprecise sensor data (GPS, temperature, ...) is easily smoothed

- Built-in window functions make smoothing really simple

4

Failure recovery and highly-available setup

Checkpointing guarantees state consistency in case of failures

- Applications are automatically recovered with exactly-once state guarantees
- Output consistency (aka end-to-end exactly-once) can be provided as well*

Resource managers restart failed Flink processes

- Kubernetes, YARN, Mesos

*Available guarantees depend on sink system

5

Define and match patterns on event streams

Define REGEX-like patterns and match them on event streams

- “Identify orders that are not completed in time”
- Java & Scala Complex Event Processing (CEP) library based on DataStream API
- Implementation of SQL:2016’s MATCH_RECOGNIZE clause

Combine pattern matching with data analytics

- “Count how often a pattern matched within 5 minutes”
- “Match pattern if an two events occurred more than 10 times within 5 minutes”

Perfect match for many IoT use cases

6

Well-connected with external systems

Large library of community maintained connectors

- Messaging systems:
  Apache Kafka, AWS Kinesis, Apache Pulsar, RabbitMQ, …
- Databases & KV Stores:
  Cassandra, Elasticsearch, JDBC
- Files:
  HDFS, S3, Parquet, ORC, Avro

7

Data streaming is (conceptually) simple

Stream processing is the natural way of handling IoT events

- Events are processed one by one

Flink’s DataStream API makes stream processing easy

- Gives access to core ingredients: State & Time
- Offers a lot of built-in operators (Windows, Joins, CEP)
- Applications can be parallelized and scaled to any size

ItemAware

An Asset Tracking System Built with Flink

EVANHOE & ASSOCIATES

Freeport Metrics | Our Background

B2B Digital products development

Strong data processing and data analytics roots

We have worked with

- Solar and wind farms data
- Inventory and warehouse systems
- Automated retail kiosks
- Sustainability reporting (e.g. carbon footprint)

Before Flink we relied mostly on a mix of traditional ETL tools and our own custom solutions

Asset Tracking System:
The Challenge

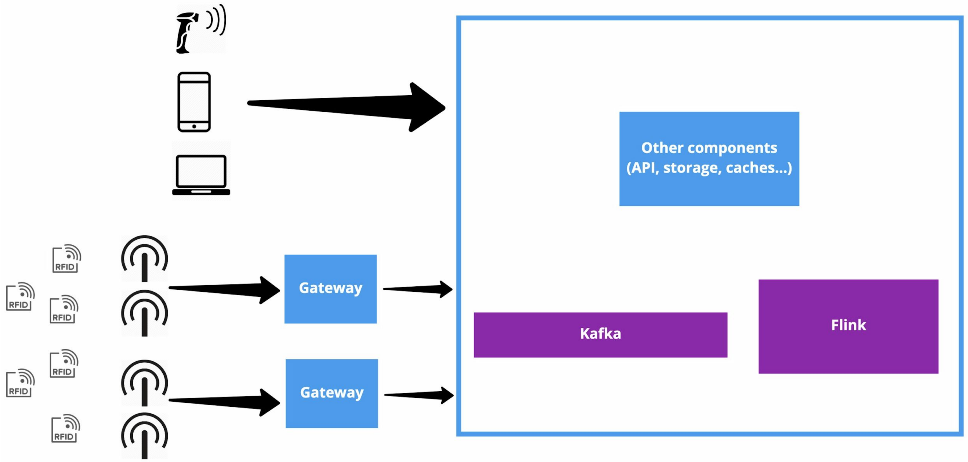
Example use cases

- Inventory management
- Tracking shipments in warehouses
- Hospital dashboards for families in waiting rooms

Multiple data sources

- RFID tags & antennas
- Hand-held barcode and RFID scanners
- User mobile and web applications

100s to 100s of thousands assets tracked in real time



Event Time

Ordering data from multiple sources by event time

- RFID Antennas
- Handheld RFID scanners
- Web and Mobile UI

Windowing - cleaning data e.g. overlapping antennas

State

Partitioning - all events for a single tag can be grouped together

Parallelizable by default - critical for performance

Complex event processing and business process modeling

- If a patient
  - entered the waiting room
  - and then entered the doctor’s office
  - and then entered the recovery area and 10 minutes passed
  - => the procedure is over

Dev Experience

We achieved our goals - functionality and performance

We could focus on logic

Flexibility - high level abstractions to low level functions

Good integrations with external tools - Kafka

Challenges

- New ‘way of thinking’ - not just another framework
- Adding new features requires careful planning for all affected parts of the system
- 3 years ago, learning materials were limited - not the case anymore

More IoT Use Cases for Flink

Data-driven agriculture
@John Deere

John Deere

- Manufacturer of machines for agriculture, construction, and forestry (Fortune 500)
- Runs a data platform to provide data services for farmers

Data-driven agriculture

- The data platform receives and processes 4 Billion geospatial events per day
  - A single planting machine produces 2400 sensor measurements per second
  - Low connectivity in rural areas, spiky data reception
- Data is rasterized, time-windowed, and stored in a data lake for later analysis & visualization

Setup

- Kinesis -> Flink -> S3 / DynamoDB
- Flink on AWS EMR

https://sf-2019.flink-forward.org/conference-program#how-john-deere-uses-flink-to-process-millions-of-sensor-measurements-per-second

Living Maps @Here

Here

- Provides mapping and location data and related services
- Open location platform with a data marketplace and Flink as a service

Building living maps

- Enhance maps with live data: slippery roads, variable signs, accidents, parking spots
- Process IoT events from car sensors: GPS, slope, wheel, side distance, sign detectors
- Flink application enriches events with location context

https://berlin-2018.flink-forward.org/conference-program#real-time-processing-of-noisy-data-from-connected-vehicles

Fleet Management @Trackunit

Photo by MGI Construction Corp. (https://flic.kr/p/EEux6C), (CC BY-ND 2.0)

Trackunit

- Provides telematics solutions and fleet management systems for the construction industry
- Offers IoT services to optimize the daily operations of its customers
- Creates both hardware and software solutions within telematics and industrial IoT

Fleet management in the construction industry

- Pipeline to process telematic IoT data
- Track locations, idleness and usage patterns, maintenance intervals of machines

Setup

- Kinesis -> Flink -> Cassandra
- Flink on AWS EMR

https://www.ververica.com/blog/trackunit-leverages-flink-industrial-iot


Bushfire Detection
@AWS blog (demo)
Photo by bertknot (https://flic.kr/p/dwVX5b), (CC BY-SA 2.0)

Not a real-world application

- Blog post describes the use case and how to run the demo on AWS

Bushfire detection

- Demo assumes a multi-hop wireless mesh network of long-lived battery powered IoT sensors
    - Sensors propagate the temperature readings to neighbors until a gateway is reached
- Demo uses Flink’s CEP library to detect bushfire patterns from incoming temperature events
- A real-time heat-map visualization shows the area under surveillance

Setup

- Kinesis -> Flink -> Amazon ES / Kibana
- Flink on AWS EMR

https://aws.amazon.com/blogs/big-data/real-time-bushfire-alerting-with-complex-event-processing-in-apache-flink-on-amazon-emr-and-iot-sensor-network/

Conclusion

Flink meets the high demands of IoT applications.

Performance & Failure Handling

- Handles latency requirements and huge data volumes
- Automatically recovers from failures and guarantees consistency

Application logic

- High-level API to ease implementation of complex logic
- Low-level API to implement any functionality
- Event-time processing to control timing issues of IoT events

Flink is being used for many IoT projects.

Try Flink for your next IoT project as well!

THANK YOU!