Introduction to Apache Flink® via SQL

Fabian Hueske – Software Engineer

The complete material of the tutorial
is available at

https://github.com/ververica/sql-training

About Me

- Apache Flink PMC member & ASF member
  - Contributing since day 1 at TU Berlin
  - Focusing on Flink’s relational APIs since ~3.5 years
- Co-author of “Stream Processing with Apache Flink”
  - Expected release: April 2019!
- Co-founder of data Artisans (now Ververica)

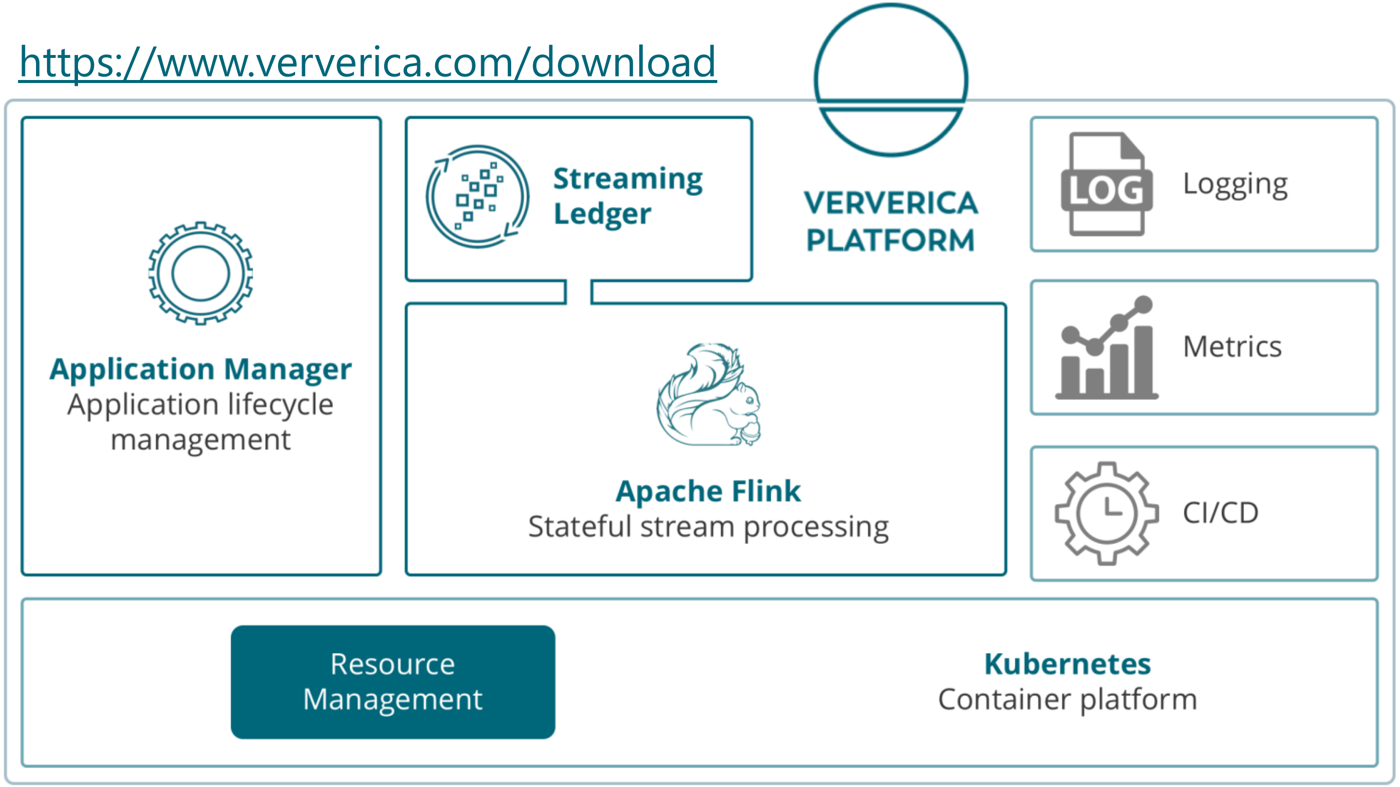
About Ververica

Original creators of Apache Flink®

Complete Stream Processing Infrastructure

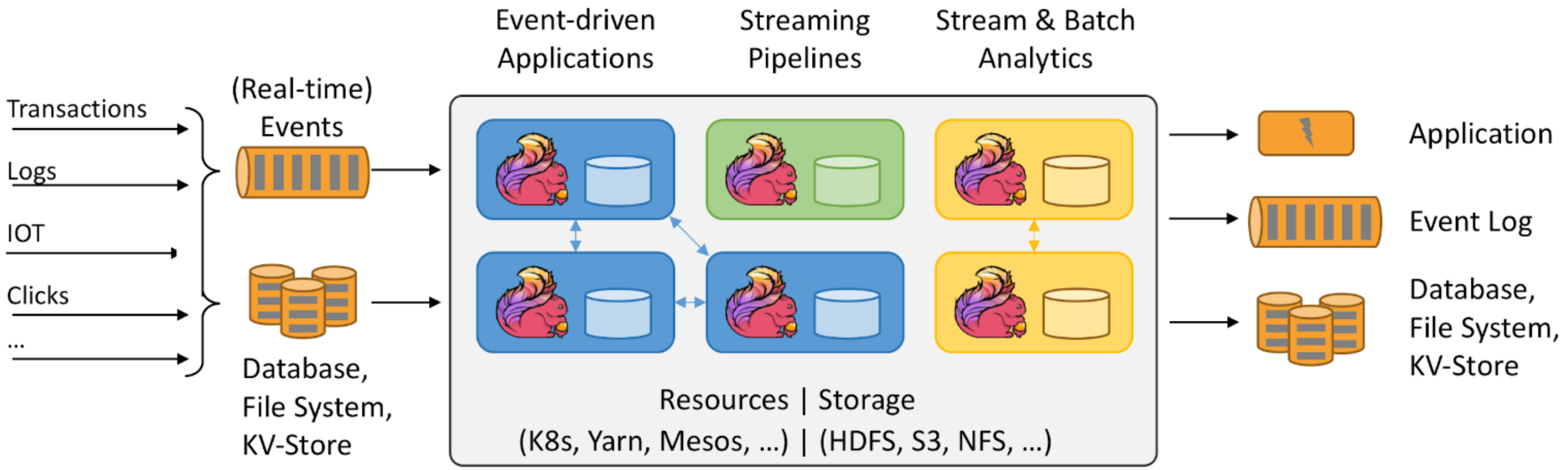
Ververica Platform

https://www.ververica.com/download



What is Apache Flink?

Stateful computations over streams
real-time and historic
fast, scalable, fault tolerant, in-memory
event time, large state, exactly-once



Event-driven Applications
Streaming Pipelines
Stream & Batch Analytics
Transactions
Logs
IOT
Clicks
...
(Real-time) Events
Database, File System, KV-Store
Resources | Storage
(K8s, Yarn, Mesos, ...) | (HDFS, S3, NFS, ...)
Application
Event Log
Database, File System, KV-Store

Hardened at Scale

UBER

Streaming Platform Service
billions messages per day
A lot of Stream SQL

NETFLIX

Streaming Platform as a Service
3700+ container running Flink,
1400+ nodes, 22k+ cores, 100s of jobs,
3 trillion events / day, 20 TB state

Alibaba Group

1000s jobs, 100.000s cores,
10 TBs state, metrics, analytics,
real time ML,
Streaming SQL as a platform

ING

Fraud detection
Streaming Analytics Platform

Powered by Apache Flink

Alibaba Group
aws
NETFLIX
UBER
HUAWEI
COMCAST

ING
cisco
Tencent 腾讯
Capital One
lyft
accenture

king
yelp
BetterCloud
Expedia
ebay
DELL EMC

MediaMath
here
Lightbend
Telefónica
D TRB
zalando

amadeus
otto group
LINE
relayr.
bouygues TELECOM
OVH.com

bol.com
SK telecom

Flink's Powerful Abstractions

Layered abstractions to
navigate simple to complex use cases

```
SELECT room, TUMBLE_END(rowtime, INTERVAL '1' HOUR), AVG(temp)
FROM sensors
GROUP BY TUMBLE(rowtime, INTERVAL '1' HOUR), room
```

High-level
Analytics API

SQL / Table API (dynamic tables)

Stream- & Batch
Data Processing

DataStream API (streams, windows)

```
val stats = stream
    .keyBy("sensor")
    .timeWindow(Time.seconds(5))
    .sum((a, b) -> a.add(b))
```

Stateful Event-
Driven Applications

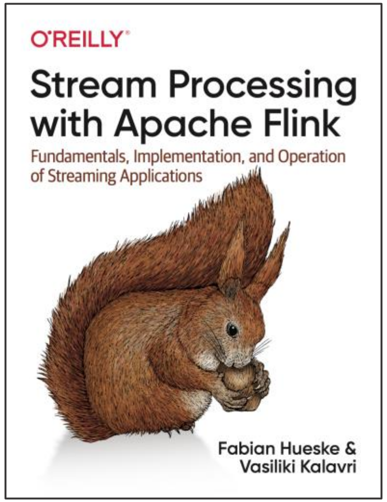
Process Function (events, state, time)

```
def processElement(event: MyEvent, ctx: Context, out: Collector[Result]) = {
  // work with event and state
  (event, state.value) match { … }

  out.collect(…) // emit events
  state.update(…) // modify state

  // schedule a timer callback
  ctx.timerService.registerEventTimeTimer(event.timestamp + 500)
}
```

Let's get started…

- This tutorial focusses on Flink SQL
  - Exercises are based on a Docker Environment
  - Please install now Docker if you haven't already.
  - http://github.com/ververica/sql-training

- Interested in Flink's low-level APIs?
  - Stream Processing with Apache Flink will be available soon!


O'REILLY

Stream Processing with Apache Flink

Fundamentals, Implementation, and Operation of Streaming Applications

Fabian Hueske & Vasiliki Kalavri

The Apache Flink® Conference

San Francisco | April 1-2, 2019

Use SquirrelSF19 for 15% off

flink-forward.org

FLINK FORWARD

Organized by ververica

#flinkforward

www.ververica.com

@VervericaData