The Fast and the Furious –

Echtzeit Analyse von Big Data Feeds mittels Flink und Kafka

Michael Schaefers

SCOOP SOFTWARE

Agenda

01 Motivation

02 Architekturbild

03 Kafka und Flink im Überblick

04 Event Processing – etwas Theorie

05 Threat Management in der Praxis

SCOOP SOFTWARE

01 Motivation

Big Data Feeds

Echtzeitanalyse

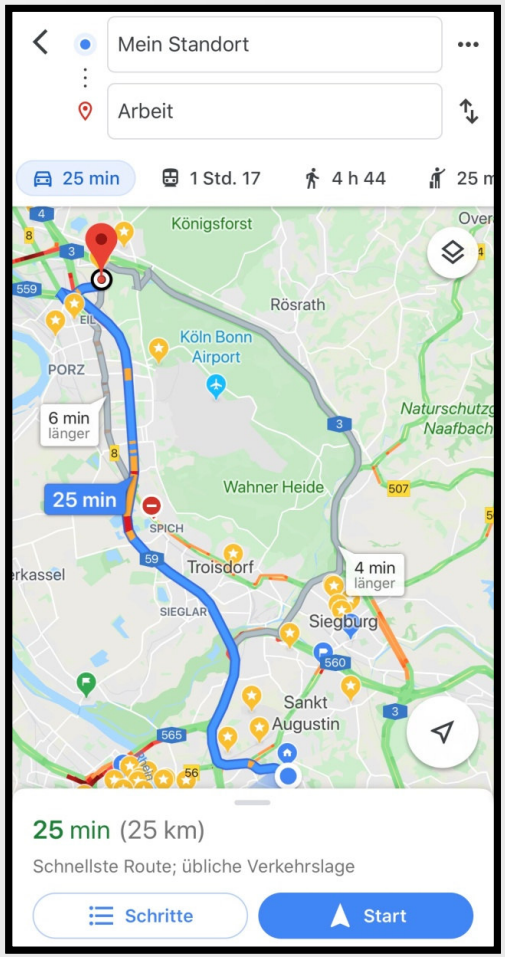
SCOOP SOFTWARE

Big Data Feeds

- Google Maps Bewegungsdaten
- IoT: Beispiel Digital Farming: Sensordaten auf Feldern
- Connected Car – Sensordaten autonomer Fahrzeuge
- Logdateien (Netzwerkdaten, Proxylogs)

Echtzeitanalyse

- Google Maps: Stauinformationen an die Endgeräte
- Netzwerkdaten: Ausfälle oder Peaks schnell erkennen
- Logdateien: Angriffe oder Malware schnell erkennen



SCOOP SOFTWARE

Echtzeitanalyse: Fragestellungen

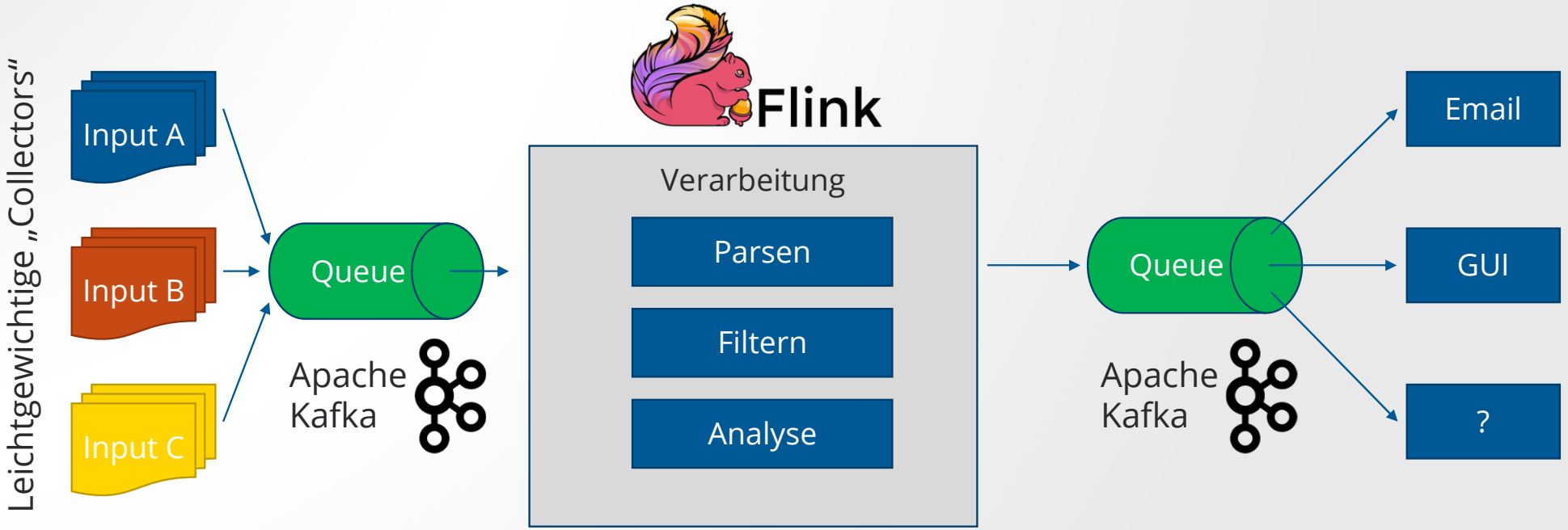
- Hat jemand eine „böse“ IP-Adresse aufgerufen?
- Versucht ein Benutzer innerhalb kurzer Zeit öfter als 10x sein Passwort zurückzusetzen?
- Hat ein Benutzer ein Word-Dokument heruntergeladen und wurden kurz danach sehr viele Internet-Zugriffe aus Word heraus ausgeführt?

SCOOP SOFTWARE

02 Architekturbild

Wie könnte ein System zur Threatanalyse aussehen?

Architekturbild



SCOOP SOFTWARE

03
Kafka & Flink

Ein Überblick

SCOOP SOFTWARE

Kafka: Übersicht

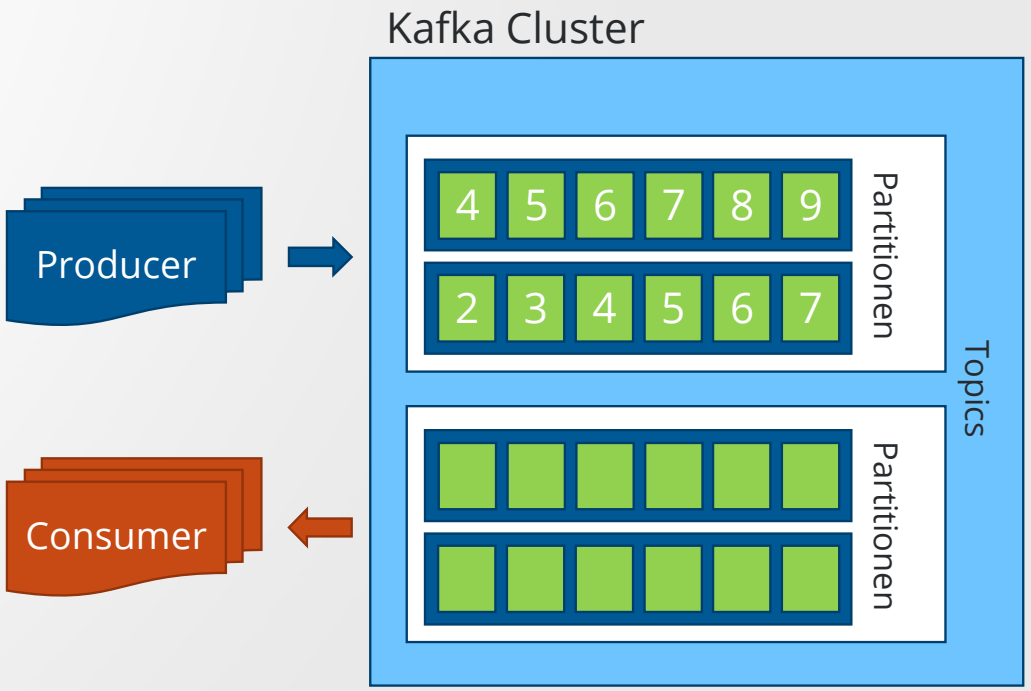
- „Distributed Streaming Platform“
- Verwaltungstool für Datenströme fast beliebiger Art
  - Event Log
  - Transaction Log
- Open Source Projekt der Apache Software Foundation
- Kommerziell von Confluent getrieben

SCOOP SOFTWARE

Kafka: Kernarchitektur

APACHE kafka
A distributed streaming platform

- Serielle Eventlogs
- Topics trennen logisch
- Partition trennen technisch
- Producer schreiben
  - immer ans Ende einer Partition
- Consumer lesen
  - ab einem bestimmten Offset
  - Offset pro Consumer & Partition
  - immer seriell
- Publisher / Subscriber Pattern



SCOOP SOFTWARE

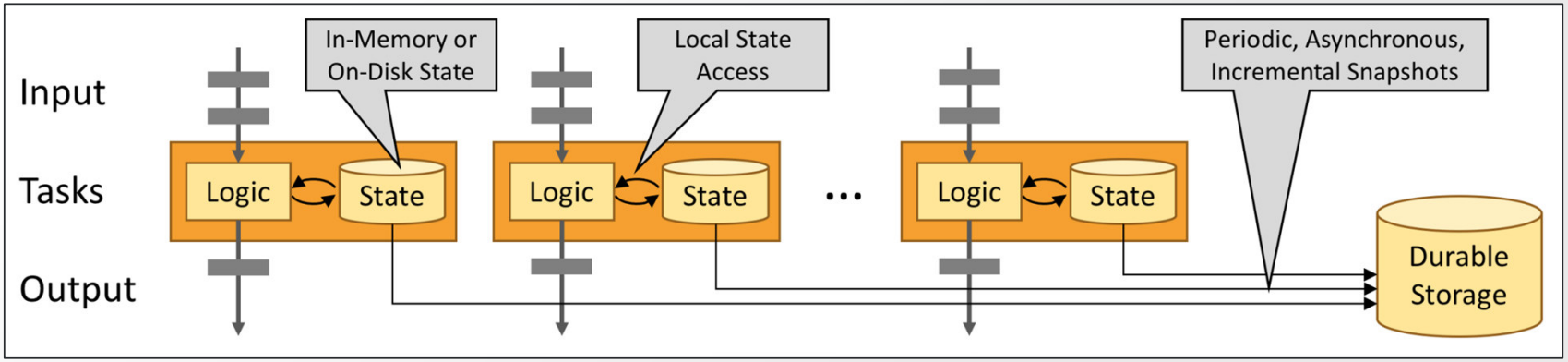
Kafka: Cloud Features

- Produktiver Einsatz immer als Rechnerverbund
- Horizontale Skalierbarkeit
  - Skaliert quasi-linear mit Ressourcen (max. mit Anzahl Partitionen)
- Kein Single Point of Failure
  - Mehrere Bootstrap-Server im Connect-String konfigurierbar
- Redundante Datenhaltung für Ausfallsicherheit
  - Jede Partition hat einen „Leader“ Host
  - Beliebig viele „Follower“ Hosts synchronisieren sich im Hintergrund
- Automatische Lastverteilung durch Broker
  - Anzahl Clients mit gleicher ID zur Laufzeit änderbar
  - Partitionszuweisung an Clients erfolgt durch Broker

SCOOP SOFTWARE

Flink: Übersicht

- “Stateful Computations over Data Streams”
- Arbeitet auf Standard Clusterumgebungen (z.B. YARN, Kubernetes)
- Quasi beliebige Skalierbarkeit
- Optimiert auf In-Memory Performance (Data Locality)
- Open Source Projekt der Apache Software Foundation

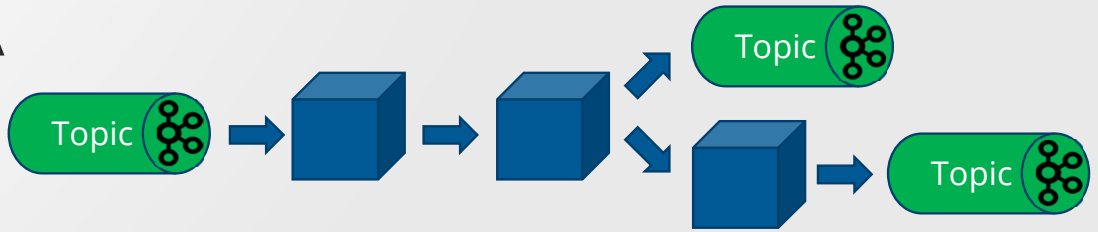


https://flink.apache.org/flink-architecture.html

SCOOP SOFTWARE

Flink: Kernarchitektur

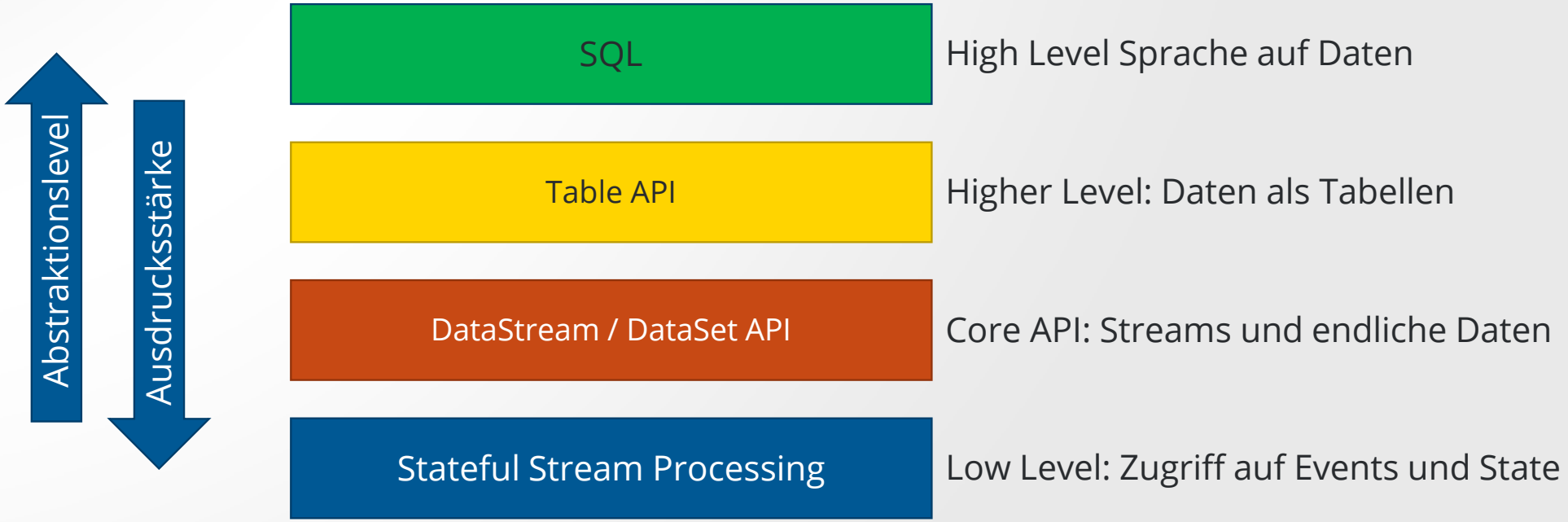
- Definition einer Processing Pipeline („Job Graph“)
  - Stream Sources (bounded / unbounded)
  - Tasks (Verarbeitungslogik, State)
  - Stream Sinks
- Ein Job definiert eine oder mehrere Pipelines
- Job Manager (mind. 1 pro Job)
  - Orchestriert Task Manager
  - Mehr als ein Job Manager für HA
- Task Manager (n pro Job)
  - Prozessiert Teile einer Pipeline
  - Parallelität pro Task möglich



SCOOP SOFTWARE

Flink: Für jeden eine API

- Mehrere APIs auf unterschiedlicher Abstraktionsebene



https://ci.apache.org/projects/flink/flink-docs-release-1.9/concepts/programming-model.html

SCOOP SOFTWARE

Flink: Abgrenzung

- Apache NiFi: Einfaches Event Processing mit Connection-Graph
  - Eher für Datenströme zwischen Anwendungen
- Kafka Streams/KSQL: einfach zu benutzen, weniger mächtig als Flink
  - Kein Complex Event Processing, keine Custom Trigger
- Spark Streaming: Hoher Bekanntheitsgrad, aber kein echtes Streaming
  - Sog. Micro-Batches statt Streaming, keine komplexen Pipelines
- Flink: Echtes und komplexes Streaming, sehr gute Layered API
  - High-Level oder Low-Level API gleichermaßen gut nutzbar
- Google Cloud DataFlow: Mächtigkeit von Flink + automatische Skalierung
  - Allerdings nur in der Google Cloud

SCOOP SOFTWARE

04
Event Processing: Theorie

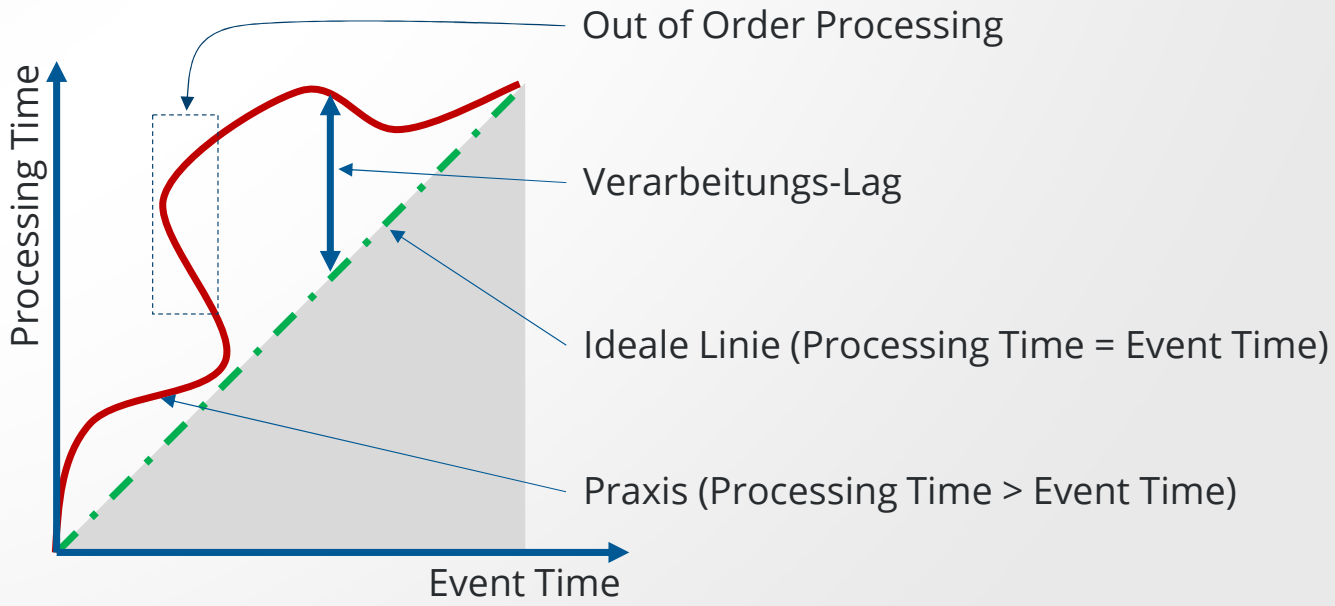
Timestamp Problematik

Windowing, Watermarks

SCOOP SOFTWARE

Event Time vs. Processing Time

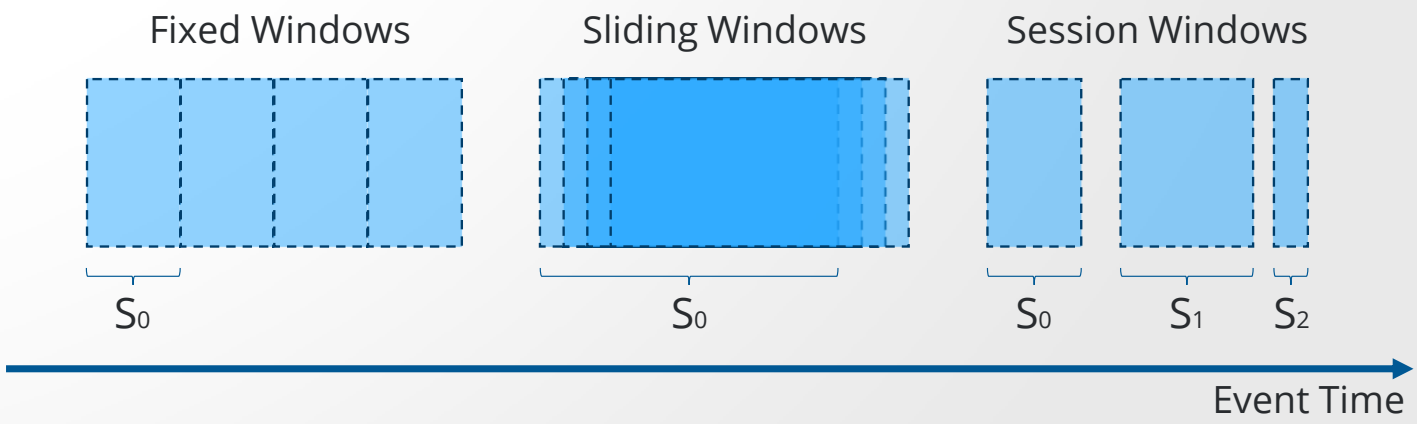
- „Wann ist das Ereignis aufgetreten“ vs. „Wann wird das Ereignis verarbeitet“
- In der Regel ist die Event Time älter
- Flink: Arbeitet auf Event Time



SCOOP SOFTWARE

Windowing

- Viele Stream Operationen arbeiten auf Zeitfenstern
  - “Anzahl Loginversuche innerhalb von 1 Minute”
- Unterschiedliche Arten von Zeitfenstern
- Out of Order Processing: Wann ist ein Zeitfenster komplett?



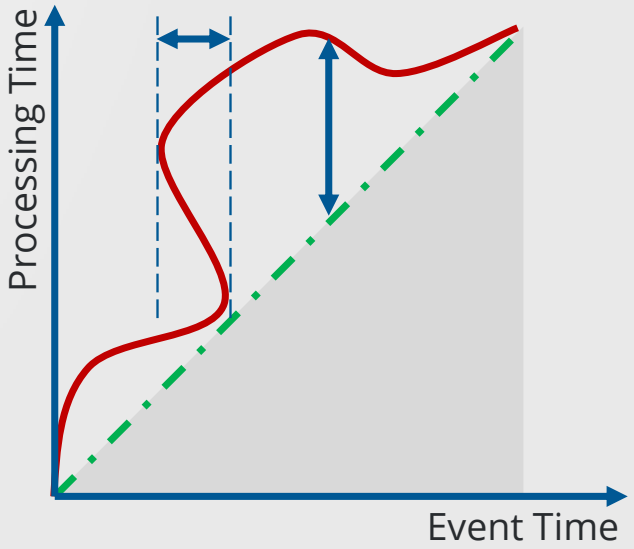
SCOOP SOFTWARE

Watermarks

- Definieren, wann ein Zeitfenster komplett ist
- Technisch: Ein Zeitpunkt. Alle Zeitfenster, die dahinterliegen, sind komplett

Beispiele:

- Erlauben eines maximalen Processing Lags
- Beschränken der maximalen Out-of-Orderness
- Flink erlaubt maximale Freiheit:
  Eigenentwicklung möglich



SCOOP SOFTWARE

05
Threat Analyse in der Praxis

Anforderungen

Tuning

SCOOP SOFTWARE

Anforderungen

- Unterschiedliche Datenformate (Proxylogs, Windows Events, AWS Logs, etc.)
- Über 1.300.000.000 Events pro Tag zu verarbeiten
- Der Großteil: Proxylogs (IP, User, Host, URL, Response Code, etc.)
- Analyseregeln per GUI definierbar
- Komplexität der Regeln in der kompletten Bandbreite
- Viele gleichzeitige Regeln aktiv ➔ Skalierbarkeit
- Regelmäßiges PDF Reporting / Offline Analyse
- Möglichst zeitnahe Analyse der Daten (minimaler Processing-Lag)
- [...]

SCOOP SOFTWARE

Tuning Proxylogs

- Proxylogdateien pro Server immer nach Event Time sortiert ➔ ausnutzen!
- Pro Server eine Kafka Partition
- Event Timestamps innerhalb einer Partition sind auch aufsteigend sortiert
- Es gibt keine Out-of-Orderness
- Dadurch keine verzögernden Watermarks

➔ Sehr viel näher an Realtime Processing

➔ Weniger Ressourcen durch wenige, schnell schließende Windows

SCOOP SOFTWARE

06
Fazit

SCOOP SOFTWARE

Fazit: Kafka und Flink - ein Dream Team

- Kafka ist State-of-the-Art für High-Performance Event-Streaming
- Flink ist extrem mächtig, gute Abstraktion durch API-Layers
- Durchaus hohe Lernkurve, insbesondere bei Flink
- Flink ist unverzichtbar bei komplexen Use Cases

> At the heart of this massive data ingestion pipeline is a self-serve stream processing platform that processes 3 trillion events and 12 PB of data every day. We have recently migrated this stream processing platform from Samza to Flink
>
> Steven Wu, NETFLIX

https://www.ververica.com/flink-forward/resources/scaling-flink-in-cloud

Kontakt

SCOOP Software GmbH
Gut Maarhausen
Eiler Straße 3P
D-51107 Köln

Michael Schaefers
michael.schaefers
@scoop-software.de
www.scoop-software.de