Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

Verspätungsanalyse via SIRI-ET API

Datengetriebene Auswertung von Verspätungen im Schweizer ÖV mit Apache Spark

Modul: Big Data CAS Information Engineering

Gruppe: BD02

Autoren: Flavio Suhner, Niels Meier, Pascal Gubler

Abgabedatum: 30. Juni 2025

Inhaltsverzeichnis

- Ziel des Projektes
- Schritt 1: Live Daten abrufen und analysieren
- Schritt 2: Übergabe der Daten an Spark
- Schritt 3: Analyse - Top 10 verspätete Linien
- Schritt 4: Visualisierung - Heatmap der Top 10
- Schritt 5: Performance Test – Vergleich mit/ohne AQE
- Abschluss & Demo

Ziel des Projektes

Ziel war es, mit Hilfe der SIRI-ET API von „opentransportdata.swiss“ die aktuell verspäteten Linien im ÖV zu identifizieren, zu analysieren und grafisch aufzubereiten.

Zusätzlich wurde ein Performancevergleich zwischen klassischer und adaptiver Verarbeitung mit Apache Spark (AQE) durchgeführt.

Schritt 1: Live Daten abrufen und analysieren

- Verbindung zur *SIRI-ET API*
- Echtzeitdaten im *XML-Format*
- Extraktion von:
  - Liniennummer
  - Fahrrichtung
  - Haltestelle
  - Geplante & erwartete Abfahrtszeit
- Berechnung der Verspätungen
- Ausschluss von Ausreissern
- Überführung in ein Pandas *DataFrame*

Schritt 2: Übergabe der Daten an Spark

- Spark-Session via ZHAW-Notebook (2 Kerne)
- Konvertierung des Pandas DataFrame in ein Spark DataFrame
- Vorteil: skalierbare Vearbeitung grosser Datenmengen
- Erste Sichtprüfung df.printSchema(), df.show()

Schritt 3: Schritt 3: Analyse Top 10

- Gruppierung nach line und direction
- Berechnung der duchschnittlichen Verspätung je Gruppe
- Sortierung nach höchster Durschnitsverspätung
- Top 10 verspätete Linien identifizieren
- Beispiel: IC Berlin Hbf, ICE Chur, Linie 3 nach Baden

Schritt 4: Schritt 3: Visualisierung

- Umwandlung in Pandas DataFrame
- Kombination Linie & Richtung zu Label
- Heatmap mit Seaborn erstellt
- Farbskala in Rottönen (je dunkler, desto verspäteter)
- Erkenntnis: einzelne Linien stark überdurchschnittlich betroffen

Schritt 5: Performance-Test (AQE)

- Vergleich verschiedener AQE-Konfigurationen in Spark:
  - AQE deaktiviert
  - Nur Join-Strategie
  - Nur Coalesce
  - Nur Skew Join
  - Alle AQE-Features aktiviert
- Je 3 Durchläufe pro Einstellung
- Darstellung im Boxplot
- Ergebnis: AQE reduziert Laufzeit teils deutlich

Abschluss

- Kombination aus Pandas, Spark und Seaborn = effektives Framework
- Öffentliche Live-Daten lassen sich effizient analysieren
- AQE-Optimierungen lohnen sich bei grossen Datenmengen

Demo