38. Proudové (streamové) zpracování dat- Spark Streaming a Structured Streaming- popis, rozdíly, výhody a nevýhody.

- komponenta Sparku určená pro analýzu streamovaných dat
  - škálovatelná s vysokou propustností a odolností vůči chybám
    - checkpointy jsou pravidelně ukládány na disk
  - konstantní přísun dat z různých zdrojů
    - např. zpracování logů z webu nebo serveru
- data jsou agregována a analyzována v určitý daný interval
- data mohou pocházet z různých zdrojů
  - Amazon Kineses, HDFS, Kafka, Flume, sockety
- opět existují dvě podobná API
  - původní založené na RDD's nazývané Spark Streaming
  - nové založené na DataFramech nazývané Structured Streaming

https://spark.apache.org/docs/latest/streaming-programming-guide.html

Spark streaming

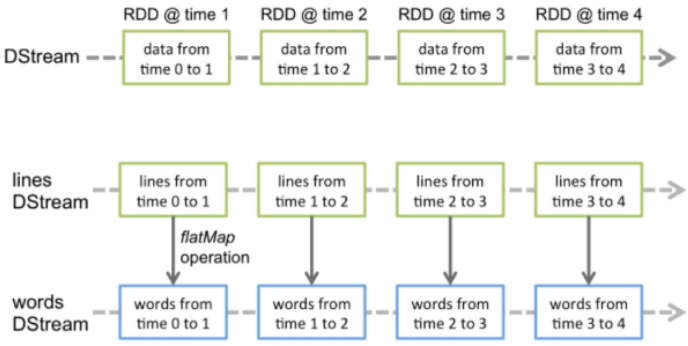
jak funguje Spark Streaming?

- přijímá data ze streamů a dělí je na dávky
- dávky jsou zpracovány Spark enginem
- výsledky jsou generovány ve formě streamu dávek

https://spark.apache.org/docs/latest/streaming-programming-guide.html

- nejedná se tedy o úplně real-time zpracování, ale micro-batching
- Spark poskytuje vysokoúrovňovou abstrakci DStream (discretized stream)
  - reprezentuje konzistentní tok dat
  - interně je reprezentován jako sekvence RDD's

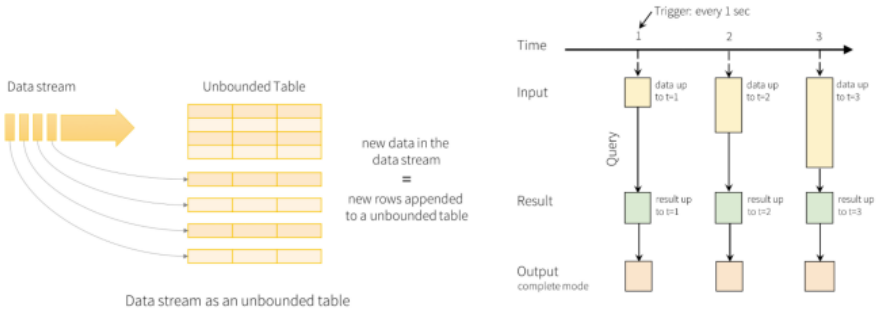
- reálná ukázka: počet slov textu na vstupu
  - připojení k lokálnímu streamu dat přes TCP socket



jak funguje Structured Streaming?

- streamovaná data jsou modelována jako DataFramy, které jsou rozšiřovány
  - umožňuje zpracování streamů stejným způsobem jako statických dat
- data jsou udržována ve vstupní tabulce
- nová data jsou jen připnuta do vstupní tabulky
- vstupní tabulka je zpracovávána, čímž je aktualizována výsledková tabulka
- po aktualizaci výsledkové tabulky dochází k zápisu ven
  - podpora různých módů
  - complete zapíše celou tabulku
  - append připíše jen nové řádky
  - update zapíše jen zaktualizované řádky
- možnost volby mezi micro-batch a continuous zpracováním
  - použití jen přepnutím módu s možností snížit latenci

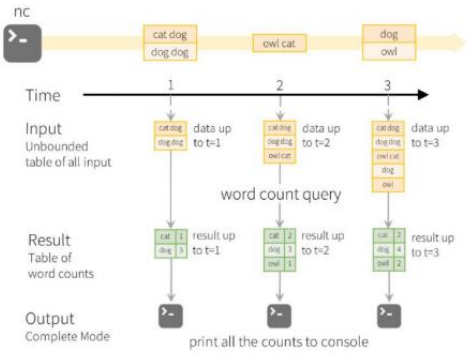
- jak funguje Structured Streaming?



Data stream as an unbounded table

Programming Model for Structured Streaming

- reálná ukázka: počet slov textu na vstupu
  - připojení k lokálnímu streamu dat přes TCP socket



Model of the Quick Example

- podporované operace
  - většina funkcionality je podporována
  - selekce, projekce, agregace
  - windowed operace
  - join operace
    - se statickými i streamovanými DataFramy
- nepodporované operace
  - vícenásobné streamované agregace
  - limit na prvních N řádků
  - distinct
  - některé outer joiny
  - řazení jen po agregaci a v complete módu

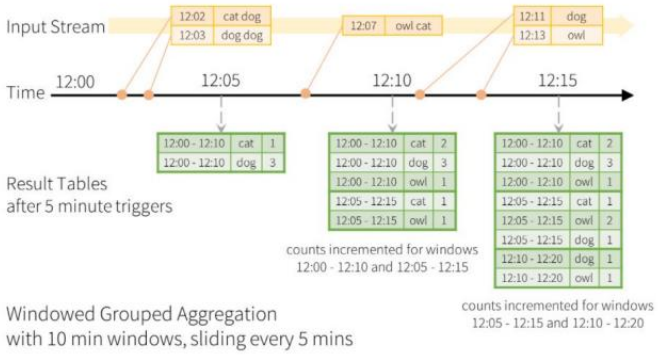
Porovnání

- Spark Streaming
  - původní API od verze 0.7
  - založené na DStreams
    - interně využívá RDD's
    - optimalizace ze strany vývojáře
  - micro-batch zpracování
  - pouze timestamp doručení dat
    - ne vygenerování
    - opožděná data mohou být problém
  - stále k dispozici a používané
    - časem pravděpodobně deprecated
  - obtížnější propojení
- Structured Streaming
  - nové API od verze 2
  - založené na DataFramech
    - případně DataSetech
    - optimalizace již zahrnuta
  - micro-batch i continuous
  - event-time koncept
    - zpracování i na základě času události
    - řeší problém opožděných dat
  - aktuální a více podporované
    - do budoucna hlavní řešení
  - jednodušší propojení

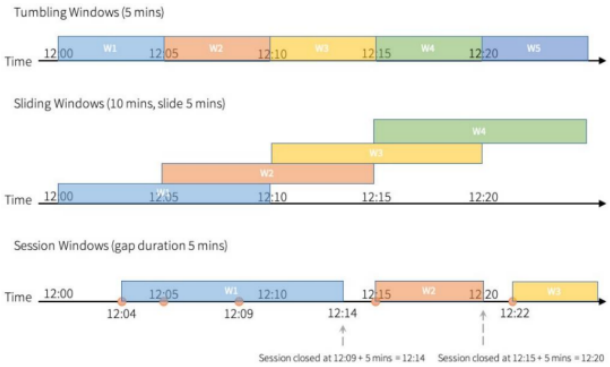
Okna a práce s nimi.

STRUCTURED STREAMING

- veškeré operace je také možné aplikovat na okna
  - možno aplikovat i na úvodní příklad čtení řádků z konzole



- veškeré operace je také možné aplikovat na okna
  - k dispozici jsou různé typy oken
  - statická (tumbling)
    - pevný interval
    - neexistují překryvy
  - posuvná (sliding)
    - pevný interval
    - existují překryvy
      - např. 5 minut
  - událostní (session)
    - různě dlouhé
    - existují pauzy



Práce s nimi

- reálná ukázka: nejčastější kódy za poslední dvě minuty
  - s posunem jedna minuta
  - rozšíření předchozí úlohy o posuvná okna
  - potřebné importy
    - Spark Session a SQL funkce jako func (bude jich použito víc)
    - time

```
from pyspark.sql import SparkSession
import pyspark.sql.functions as func
import time
```

  - vytvoření Spark Session objektu

```
spark = SparkSession.builder.appName("StreamingCodes").getOrCreate()
```

  - nastavení monitoringu adresáře, kam se ukládají logy

```
accessLines = spark.readStream.text("streaming/data")
```

- reálná ukázka: nejčastější kódy za poslední dvě minuty
  - přidání sloupce eventTime s aktuální časovou značkou
    - logy jsou staré a nemají tak v sobě informace pro posuvné zpracování
    - v reálné aplikaci by tato informace byla k dispozici z logů

```
logs = logsNT.withColumn("eventTime", func.current_timestamp())
```

  - nastavení seskupení dat a posuvného okna
    - shlukovat se bude stále podle kódu (status)
    - okno je vytvářeno nad nově přidanou časovou značkou
    - okno trvá dvě minuty s posunem jedna minuta

```
statusCounts = logs.groupBy(func.window(func.col("eventTime"), "2 minutes", "1 minutes"), func.col("status")).count()
```

  - spuštění streamovaného zpracování

```
query = (statusCounts.writeStream.outputMode("complete").format("memory").queryName("counts").start())
```