Week1_Spark_for_Data_Engineering

Spark Structured Streaming

What is streaming data

Data that is continupusly generated, and often originates from more than one source.
It is unavailable as a complete data set, and requires incremental processing

Apache Spark Structured Streaming:

- Uses Spark SQL
- Uses the same DataFrame and Dataset APIs
- Processes data in Micro-batches or continuously
- Optimizes queries using Spark SQL

Common Structured Streaming terms

- Source: The data origination location
- Sink: The location of the output
- Event time: The record creation time
- End-to-end latency: The measurement of the time needed for the data to go from source to sink
- Watermarking: Manages late arriving data

Streaming Data Operations
Apache Spark Structured Streaming:

- Performs Standard SQL operations including **select, projection, and aggregation**
- Enables window operations over event time - sliding windows with aggregation
- Supports **join** operations - joins with static DataFrames or other streams

Output modes
Output modes specify how data is written to the sink

- Append - Only new rows added
- Complete - Entire result table
- Update - Only updated rows

Streaming data sinks
Supported sinks:

- Files

    - Output files to a directory
- Kafka
    - Outputs to a kafka topics
- Foreach & Foreachbatch
    - Applies a function to each record or batch
- Console & Memory
    - Used for debugging

Monitoring and Checkpointing

- Monitor your data via Spark external listeners, which work both through external frameworks or programmatically
- Checkpointing recovers query progress on failure (set checkpoint location on HDFS)

GraphFrames on Apache Spark

Graph Theory
Graph theory is the mathematical study of modeling pairwise relationships between objects

Graphs consist of

- Vertices
- Edges that connect one vertex to another

Graphs can be either directed or undirected

- Direct graphs:
    - contain edges with a single direction between two vertices
    - Examples: manufacturing, optimization, project scheduling, Train and airline route analysis, traffic recommendations, and others
- Undirected graphs:
    - Contain edges with no defined directions
    - Examples: social relationship analysis, marketing analysis, genomic analysis, knowledge repositories, and others

GraphFrames - What is it?

An Apache Spark extension for graph processing

- Based on Spark DataFrames
- Runs queries on graphs of vertices and edges and represents data

- Contains built-in algorithms
- Exists as a separate, downloadable package

Using GraphFrames

- Runs SQL queries on vertex and edge DataFrames
- Requires that you to set a directory for checkpoints
- Performs Motif finding, which searches the graph for structural patterns

Supported Graph Algorithms

- Breadth-first search (BFS)
- Connected components (strongly connected components)
- Label Propagation Algorithm (LPA)
- PageRank
    - Developed by Google to measure importanc eand rank web pages for their search engine
- Shortest paths
- Triangle count

Supported types of data

- Ideal for modeling data with connecting relationships
- Computes relationship stranght and direction

ETL Workloads

What is ETL
ETL describes the process of moving data from a source to another destination that has different data format or structure

- Extract obtains data from a source
- Transform the data in the needed output format
- Load the data into a database, data warehouse or other storage

ETL with Apache Spark
Offers the following ETL advantage:

- Provides a well-supported big data ecosystem
- Can easily load and save popular big data sources
- Scales easily to handle large workloads

Extracting Data
Extracts data from one or more different sources. Sparks supports HDFS & the following data sources:

- Parquet
- Apache ORC
- Hive
- JDBC
  `df = read.parquet("people.parquet")`

Transforming Data

- Cleans the data
- Transforms data format to make the data more accessible for analysis
- Joins DataFrames
- Groups and aggregates data
- Uses Spark SQL operations to Select and others

```
// create view of data for SQL queries
df.createOrReplaceTempView("people")

// Use Spark SQL to clean and transform data
names = spark.sql("SELECT name FROM people WHERE age BETWEEN 13 AND 19")
```

Loading data
Loads data into data warehouse, database or other data sink
Uses available Spark data sources

Summary

Streaming data is continuously generated and often originates from more than one source, and unavailable as a complete data set. Data is continuously created; thus, past data is often unavailable because the data volume is too much to store.

Apache Spark Structured Streaming processes data streams using the Spark SQL engine and DataFrame or Dataset APIs.

Output modes of a stream append only new rows, complete the entire result table, and update only the updated rows to Files, Kafka, Foreach & ForeachBatch, and Console and Memory locations.

Spark Structured Streaming supports standard SQL operations, window on event time, and joins with static or streaming DataFrames.

Spark streaming supports external listeners, which work both through external frameworks or programmatically, act as monitors on data streams to trigger events.

Checkpointing compensates for node failures by recovering query progress on failure and enables writing streams to disk.

GraphFrames enables graph processing within Apache Spark using DataFrames. GraphFrames provides one DataFrame for graph vertices and one DataFrame for edges for use with SparkSQL. GraphFrames includes popular built-in graph algorithms for use with both edge and vertex DataFrames.

ETL is the process of extracting data from a source, transforming data into a new format, and loading data into a database, data warehouse, or other storage. Spark can extract data from multiple supported sources. You can apply the SELECT and other SQL commands for data transformations and then load the data to new repositories.