
datapath
Procesamiento
Incremental de datos
www.datapath.ai

datapath

Structured Streaming

www.datapath.ai

Data Stream

- Es cualquier conjunto de datos que crece a lo largo del tiempo. Ejemplo: JSON log.
- Los nuevos archivos de arribo se almacenan en la nube.
- Actualizan una base de datos capturada en una ingesta CDC.
- Eventos encolados en una ingesta pub/sub de mensajes Ejemplo: Apache Kafka.

Procesamiento de Stream de datos

2 Enfoques:

1. Reproceso de todo el origen de datos cada vez.
2. Sólo procesar aquella nueva data agregada desde la última actualización.

   (Structured Streaming)

Spark Structured Streaming

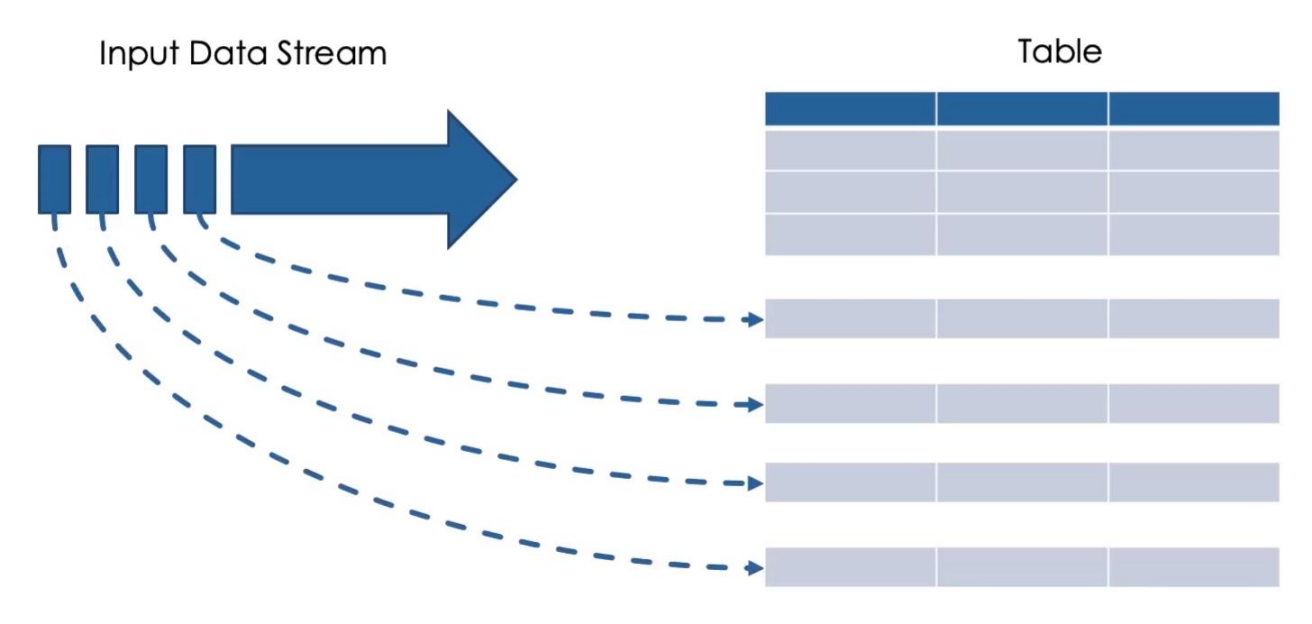
2 Enfoques:

1. Reproceso de todo el origen de datos cada vez.
2. Sólo procesar aquella nueva data agregada desde la última actualización.
   (Structured Streaming)

datapath

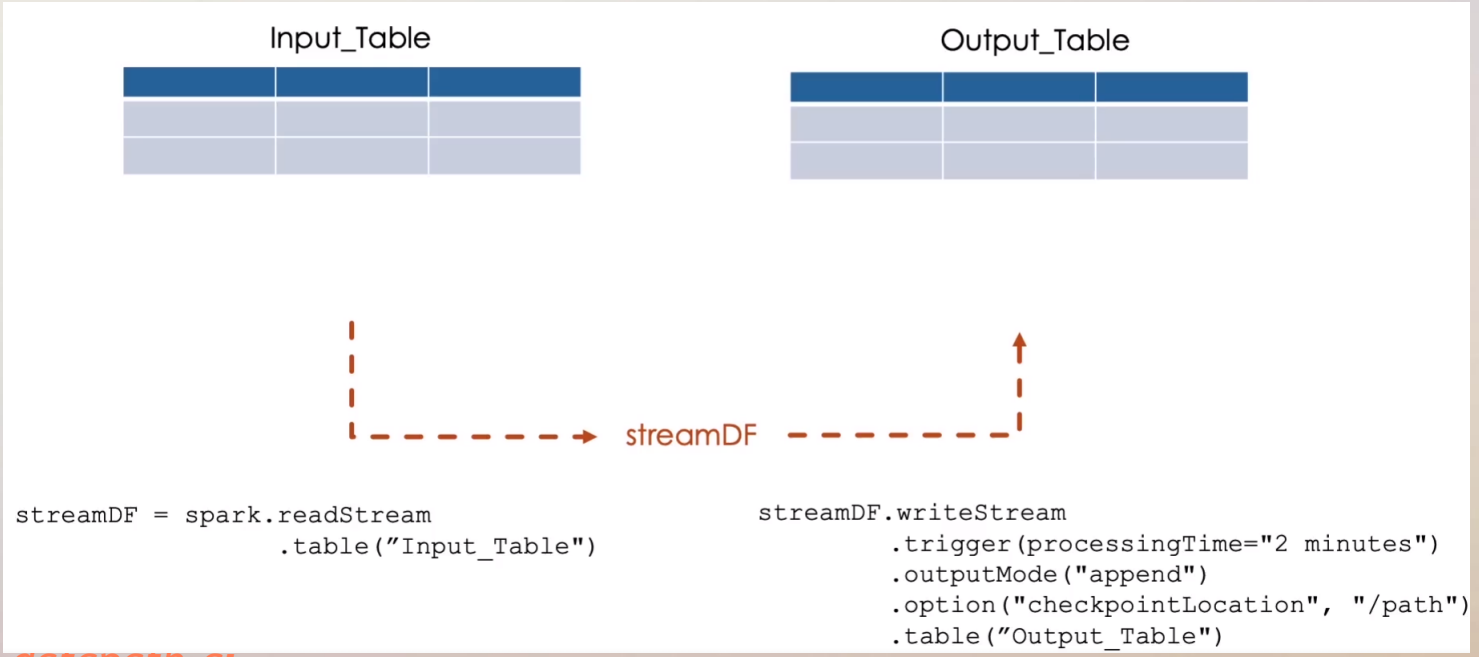
Problemas resueltos por ACID

Tratando los datos como tablas



www.datapath.ai

Tratando los datos como tablas



Intervalos de Triggers

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table("Output_Table")
```

Trigger	Method call	Behavior
Unspecified		Default: processingTime="500ms"
Fixed interval	`.trigger(processingTime="5 minutes")`	Process data in micro-batches at the user-specified intervals
Triggered batch	`.trigger(once=True)`	Process all available data in a single batch, then stop
Triggered micro-batches	`.trigger(availableNow=True)`	Process all available data in multiple micro-batches, then stop

Output Modes

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table("Output_Table")
```

Mode	Method call	Behavior
Append (Default)	`.outputMode("append")`	Only newly appended rows are incrementally appended to the target table with each batch
Complete	`.outputMode("complete")`	The target table is overwritten with each batch

Checkpointing

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table("Output_Table")
```

- Store stream state
- Track the progress of your stream processing
- Can **Not** be shared between separate streams

Guarantees

1. Fault Tolerance
   - Checkpointing + Write-ahead logs
     - record the offset range of data being processed during each trigger interval.
2. Exactly-once guarantee
   - Idempotent sinks

Operaciones no soportadas

Operaciones no soportadas por Streaming Data Frame.

- Sorting
- Deduplication

Methodos avanzads

- Windowing
- Watermarking.

Laboratorio

datapath

Ingesta incremental de archivos de datos

www.datapath.ai

Ingesta Incremental de datos

Ingestar nuevos archivos de datos desde la última ingesta.

Reducir el procesamiento redundante.

2 mecanismos.

1) **COPY INTO**
2) **AUTO LOADER**

COPY INTO

Comando SQL

Cada vez que el comando sea llamado solo cargará los nuevos archivos.

```
COPY INTO my_table
FROM '/path/to/files'
FILEFORMAT = <format>
FORMAT_OPTIONS (<format options>)
COPY_OPTIONS (<copy options>);
```

```
COPY INTO my_table
FROM '/path/to/files'
FILEFORMAT = CSV
FORMAT_OPTIONS ('delimiter' = '|',
                'header' = 'true')
COPY_OPTIONS ('mergeSchema' = 'true')
```

www.datapath.ai

AUTO LOADER

Usa Structured Streaming en Spark

Puede procesar billones de archivos.

Soporta la ingesta near real-time de millones de archivos por hora.

```
COPY INTO my_table
FROM '/path/to/files'
FILEFORMAT = <format>
FORMAT_OPTIONS (<format options>)
COPY_OPTIONS (<copy options>);
```

```
COPY INTO my_table
FROM '/path/to/files'
FILEFORMAT = CSV
FORMAT_OPTIONS ('delimiter' = '|',
                'header' = 'true')
COPY_OPTIONS ('mergeSchema' = 'true')
```

www.datapath.ai

AUTO LOADER CHECKPOINTING

Amacena la metadata de los archivos descubiertos.

Procesamiento por única vez

Tolerancia a fallos.

```
spark.readStream
        .format("cloudFiles")
        .option("cloudFiles.format", <source_format>)
        .load('/path/to/files')
    .writeStream
        .option("checkpointLocation", <checkpoint_directory>)
        .table(<table_name>)
```

AUTO LOADER + SCHEMA

```
spark.readStream
        .format("cloudFiles")
        .option("cloudFiles.format", <source_format>)
        .option("cloudFiles.schemaLocation", <schema_directory>)
        .load('/path/to/files')
    .writeStream
        .option("checkpointLocation", <checkpoint_directory>)
        .option("mergeSchema", "true")
        .table(<table_name>)
```

www.datapath.ai

datapath

COPY INTO VS AUTO LOADER

COPY INTO	Auto Loader
▶ Thousands of files	▶ Millions of files
▶ Less efficient at scale	▶ Efficient at scale

Laboratorio

datapath

Multi-Hop Architecture

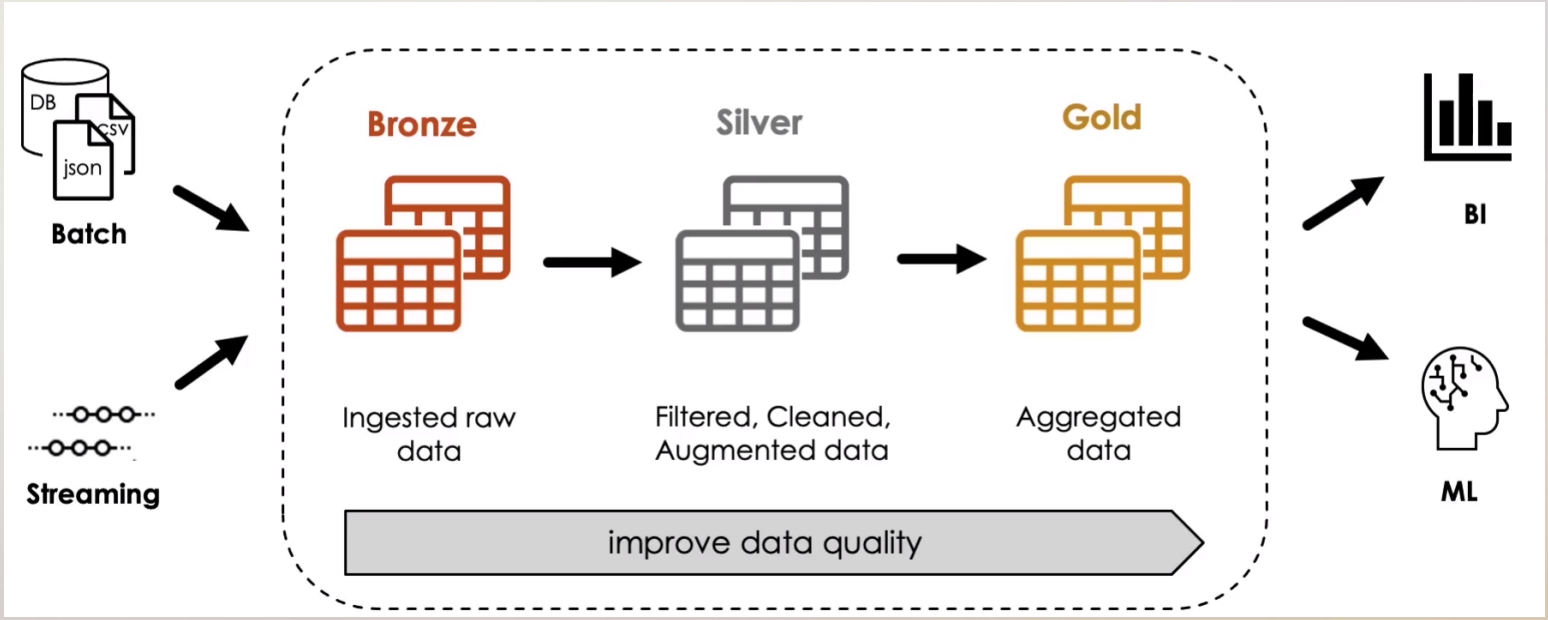
www.datapath.ai

Arquitectura Multi-Hop

- Arquitectura Medallion
- Organiza los datos en un enfoque multicapa
- Mejora de manera incremental la estructura y calidad de los datos conforme fluyan a través de cada capa.

datapath

Arquitectura Multi-Hop



www.datapath.ai

datapath

¿PREGUNTAS?

Aprende, aplica y crece

www.datapath.ai