

Ecosistema Spark

Curso académico 2017/18

Structured
Streaming

Advanced
Analytics

Libraries &
Ecosystem

Structured APIs

Datasets


DataFrames

SQL

Low level APIs

RDDs

Distributed Variables



Spark Streaming

Aplicaciones Streaming

- "Stream Processing": incorporar nuevos datos continuamente al cómputo de un resultado
- Los datos de entrada no tienen principio ni fin, es un flujo continuo de eventos
- El procesamiento consiste en computar una o varias "queries" sobre ese flujo continuo de eventos
- Los resultados de las queries se irán actualizando con el tiempo

Spark Streaming

- Requerimiento cada vez más común en apps Big Data
- Spark incluye una API de Streaming desde 2012 (Spark Streaming y la API DStreams basada en RDDs)
- API Dstreams basada en operaciones de bajo nivel contra objetos Java/Python (menos posibilidades de optimización)
- En 2016 Spark incluye "Structured Streaming", nueva API de streaming integrada con DataFrames/Datasets

Structured Streaming

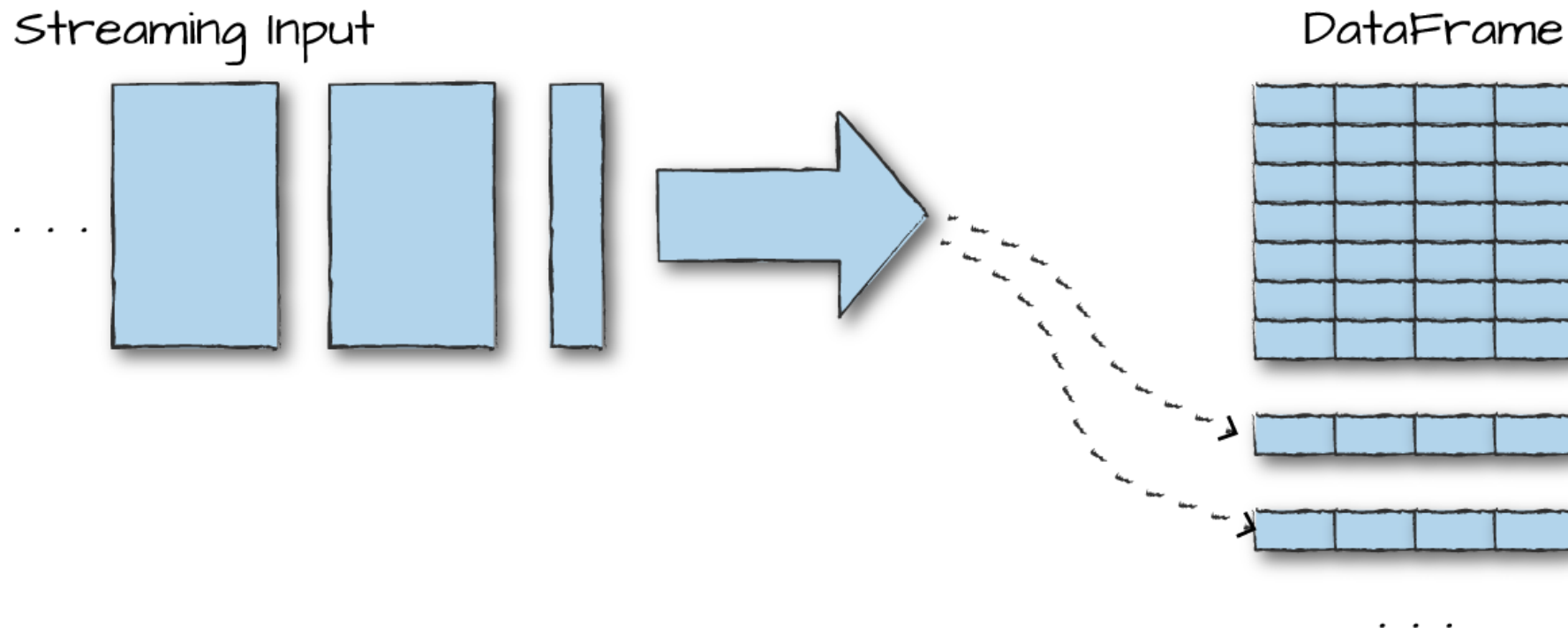
- Integrado con APIs DataFrames/Datasets
- Fácil integración con aplicaciones Spark batch
- Soporta "event time" (DStreams no lo soportaba)
- Ejemplos de uso
 - Notificaciones y alertas
 - Reports en tiempo real
 - ETL incremental
 - Online Machine Learning

Structured Streaming

- Ventajas de Streaming:
 - Menor Latencia
 - Actualización más eficiente de un resultado (incrementalmente)
- Desventajas de Streaming:
 - Mayor complejidad que los procesamientos "batch"
 - Los eventos pueden llegar desordenados
 - Es más complejo mantener mucha información de estado
 - Es más complejo no duplicar ni procesar más de una vez el mismo evento
 - Es más complejo hacer joins con datos externos
 - ...

Structured Streaming

- Trata un flujo de datos como una Tabla, a la que se están añadiendo registros continuamente



Structured Streaming

- El job de streaming verifica periódicamente si hay nuevos datos de entrada, los procesa, actualiza su estado interno, y actualiza el resultado.
- El tipo de código será el mismo que en una aplicación batch, con algunas limitaciones.
- También hay ***transformaciones*** y ***acciones***
 - Transformaciones: las mismas que en “batch”, con algunas excepciones
 - Acciones: Normalmente, se usa una sola acción, que inicia la computación continua del stream y la escritura de resultados

Structured Streaming

- Fuentes de datos:
 - Kafka
 - Ficheros en HDFS ó S3
 - Socket
- Destinos de datos (sinks):
 - Kafka
 - Ficheros
 - Consola
 - Memoria

Structured Streaming

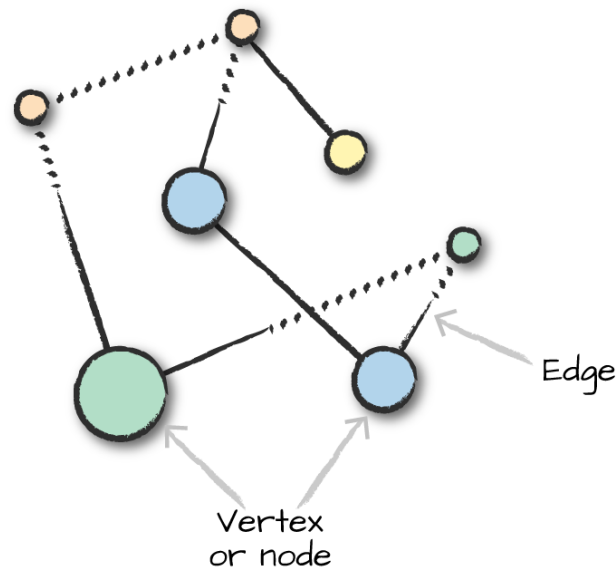
- Modos de salida: definen **cómo** queremos que Spark escriba la salida en el "sink" correspondiente.
 - **Append**: añadir sólo nuevos registros al sink de salida
 - **Update**: actualizar registros en el sink de salida
 - **Complete**: reescribir la salida completa
- No todos los sinks soportan todos los modos de salida.

GraphX



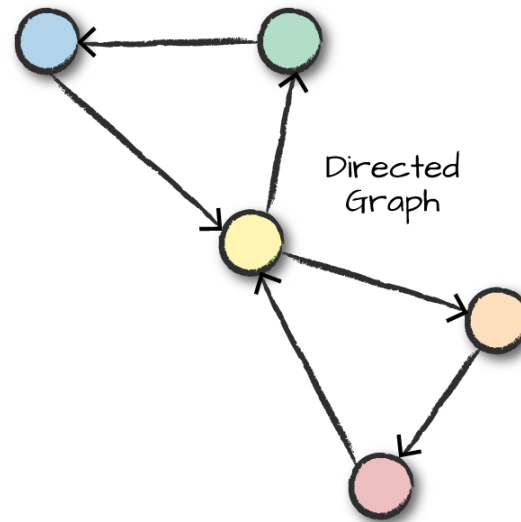
Grafos

- Estructura de datos que consiste en:
 - Nodos ó Vértices
 - Enlaces ó Aristas



Grafos

- Gran número de problemas se pueden modelizar con grafos
- Pueden ser *dirigidos* o *no dirigidos*
- El procesamiento de grafos muy grandes es complejo



Grafos

- Tanto los nodos como las artistas pueden tener datos asociados (por ejemplo, las aristas pueden tener un “peso” y los nodos un campo “nombre”)
- Múltiples aplicaciones:
 - Determinar importancia de nodos en una red (pagerank)
 - Determinar comunidades de usuarios similares (community detection)
 - Encontrar patrones en redes de comunicaciones

Graphx / GraphFrames

- Spark incluye una librería de procesamiento de grafos a gran escala, basada en RDDs: GraphX
- Existe una versión más moderna, basada en DataFrames: GraphFrames (aún no incorporada al core de Spark)
- GraphX/GraphFrames no son bases de datos de grafo, sino motores de procesamiento de grafo a gran escala.