Componentes clave de Spark en un escenario real

Caso de uso

Detección de fraude en tiempo real en transacciones financieras.
Los bancos reciben miles de transacciones por segundo y necesitan un sistema capaz de analizarlas en milisegundos para identificar patrones sospechosos y bloquear fraudes inmediatamente.

Componentes de Spark utilizados

- Driver: Coordina el proceso, recibe los datos de entrada y define el flujo de trabajo (DAG).
- Cluster Manager (YARN, Mesos o Standalone): Asigna recursos en el clúster y administra los nodos disponibles.
- Executors: Ejecutan las tareas en paralelo sobre distintas particiones de datos (cada transacción).
- RDDs y DataFrames: Procesan grandes volúmenes de transacciones en memoria y aplican transformaciones rápidas.
- Spark Streaming (o Structured Streaming): Procesa las transacciones en tiempo real desde fuentes como Kafka.
- MLlib: Aplica modelos de Machine Learning previamente entrenados (ejemplo: clasificación binaria de transacción legítima/fraudulenta).

Flujo del sistema

1. Adquisición de datos: Kafka recibe las transacciones en tiempo real desde cajeros, apps y POS.
2. Procesamiento distribuido en Spark: Spark Streaming consume los datos, los divide en micro-batches o streams continuos y los distribuye entre los executors.
3. Aplicación del modelo de ML: MLlib predice la probabilidad de fraude en cada transacción.
4. Resultados: Transacciones sospechosas se almacenan en HDFS o Cassandra y se envían alertas en tiempo real.

Representación gráfica

Usuarios → Kafka → Spark Streaming → DataFrames → MLlib (modelo fraude) → Driver coordina → Executors procesan → Resultados (HDFS + alertas)

¿Por qué usar Spark y no otra herramienta?

- Spark vs Hadoop: Spark es mucho más rápido (procesa en memoria, Hadoop depende de disco).
- Spark vs Flink: Spark tiene un ecosistema más maduro e integrado (Streaming + MLlib + SQL + GraphX).
- Spark + Kafka: combinación ideal para análisis en tiempo real a gran escala.

Conclusión

Spark permite detectar fraudes en segundos gracias a su procesamiento distribuido en memoria, integrando streaming y machine learning en un solo ecosistema.