**Procesamiento de datos en Big Data: Batch, Micro-Batch y Streaming**

El procesamiento de datos a gran escala en Big Data se divide en tres enfoques principales: **batch** (por lotes), **micro-batch** y **streaming**. Cada uno tiene características, ventajas y aplicaciones distintas.

**1. Procesamiento Batch**

El procesamiento batch implica la ejecución de tareas en grandes volúmenes de datos que se recolectan, almacenan y procesan en lotes. Generalmente, se utiliza cuando la inmediatez no es crítica, y la prioridad es procesar grandes cantidades de datos con eficiencia.

**Ejemplos de usos:**

* **Análisis de registros:** Procesamiento de logs de servidores.
* **Cálculo de informes periódicos:** Informes financieros o de ventas diarios o mensuales.
* **Procesamiento de datos ETL:** Extracción, Transformación y Carga de datos para su posterior tratamiento y análisis.

**Aplicaciones principales:**

* **Hadoop:** Framework de código abierto para procesamiento de datos masivo distribuido.
* **Apache Spark (modo batch):** Procesamiento en memoria de grandes volúmenes de datos.
* **\*\*MapReduce:** Modelo de programación que permite dividir grandes conjuntos de datos en tareas más pequeñas.

**Empresas que lo usan:**

* **Facebook (Hadoop/MapReduce -> Spark):** Para generar informes de uso y análisis de tendencias.
* **Netflix:** Procesamiento de datos históricos para recomendaciones.
* **Amazon:** Gestión de inventarios y análisis de patrones de compra.

**2. Procesamiento Micro-Batch**

El micro-batching es una variante del procesamiento batch donde los datos se dividen en pequeños lotes procesados en intervalos muy cortos (segundos o minutos). Combina elementos del procesamiento batch y del streaming.

**Ejemplos de usos:**

* **Análisis casi en tiempo real:** Revisión periódica de datos transaccionales.
* **Monitoreo de sistemas:** Recopilación de métricas en intervalos cortos.
* **Actualización de dashboards:** Datos frescos cada pocos segundos o minutos.

**Aplicaciones principales:**

* **Apache Spark Streaming:** Procesa datos en micro-lotes, ofreciendo procesamiento casi en tiempo real.
* **Azure Stream Analytics:** Para analizar flujos de datos en micro-lotes.

**Empresas que lo usan:**

* **Uber:** Para procesar datos de viajes y estimaciones de tarifas.
* **LinkedIn:** Procesamiento de notificaciones y actualizaciones.
* **Airbnb:** Análisis de disponibilidad de propiedades en intervalos cortos.

**3. Procesamiento Streaming**

El procesamiento de streaming se refiere al análisis continuo y en tiempo real de flujos de datos que llegan de manera constante. Es ideal para aplicaciones donde cada evento debe ser procesado de inmediato.

**Ejemplos de usos:**

* **Detección de fraudes:** Análisis en tiempo real de transacciones bancarias.
* **Monitoreo de redes sociales:** Seguimiento en vivo de menciones o tendencias.
* **Procesamiento de IoT:** Análisis en tiempo real de datos de sensores.

**Aplicaciones principales:**

* **Apache Kafka:** Plataforma de streaming distribuido.
* **Apache Flink:** Framework para procesamiento de flujos de datos.
* **Google Cloud Dataflow:** Procesamiento de datos en streaming en la nube.

**Empresas que lo usan:**

* **Twitter:** Procesamiento en tiempo real de tweets.
* **Tesla:** Monitoreo en tiempo real de datos de vehículos.
* **Spotify:** Recomendaciones basadas en las canciones que los usuarios escuchan en tiempo real.

**4. Conclusiones**

Dependiendo de la frecuencia de actualización de los datos, volumen y complejidad de procesamiento, optaremos por una opción u otra en función de la mayor optimización e infraestructura que dispongamos para el proyecto que tengamos en nuestras manos.

En el hilo de la compañera “Karolin Yessenia Castañeda Vásquez” viene muy bien resumido, estoy muy de acuerdo con la “tabla resumen” que comenta.

Por otro lado, también comentar que de menor a mayor coste económico para la empresa/proyecto el orden sería el siguiente “Batch -> micro-batch -> Streaming”.

**Algunas referencias:**

1. [Big Data Processing Types](https://www.linkedin.com/pulse/c%C3%B3mo-es-la-infraestructura-de-datos-facebook-mariano-minoli/%20Big%20Data%20Processing%20Types)
2. [Micro-Batching vs. Streaming](https://rivery.io/blog/batch-vs-stream-processing-pros-and-cons-2/" \t "_new)
3. [Processing with Apache Kafka and Flink](https://medium.com/@byanalytixlabs/flink-vs-kafka-a-quick-guide-to-stream-processing-engines-b09dd0e6b8af" \t "_new)