ACTIVIDAD 3 MÓDULO 8

1. Diferencia Esencial entre Procesamiento en Tiempo Real y Procesamiento Batch

La diferencia esencial entre ambos tipos de procesamiento radica en la latencia y la naturaleza del flujo de datos.

- Procesamiento de Datos Batch

  Se basa en la recopilación y el procesamiento de grandes volúmenes de datos en grupos o lotes. Este proceso se ejecuta en intervalos programados, cada hora, una vez al día o semanalmente.

  La característica principal es su alta latencia, ya que los datos no están disponibles para el análisis inmediatamente después de su creación, sino sólo después de que el lote se haya procesado completamente. Esto es ideal para tareas que no requieren una respuesta inmediata, como la generación de informes mensuales o la consolidación de datos históricos.

- Procesamiento de Datos en Tiempo Real (Streaming)

  Implica el procesamiento continuo de datos a medida que se generan, los datos fluyen a través del sistema y se procesan casi instantáneamente.

  La característica principal es su baja latencia, lo que permite tomar decisiones en tiempo real. Este enfoque es crucial para aplicaciones que necesitan respuestas inmediatas, como la detección de fraudes en transacciones bancarias o el monitoreo de equipos IoT.

Es decir, el procesamiento batch prioriza la eficiencia en el manejo de grandes volúmenes de datos, mientras que el procesamiento en tiempo real prioriza la velocidad y la inmediatez.

2. Beneficios de Usar Apache NiFi en el Proceso de Ingesta Batch

Apache NiFi es una herramienta poderosa que aporta múltiples beneficios a los procesos de ingesta de datos batch:

- Interfaz Gráfica Intuitiva

  NiFi utiliza una interfaz de usuario visual para diseñar, construir y gestionar flujos de datos. Esto simplifica el proceso de creación de pipelines, ya que no se requiere una codificación extensiva.

  Los ingenieros de datos pueden arrastrar y soltar componentes para crear flujos complejos de manera lógica y visible.

- Gestión de Flujos Robusta

  NiFi está diseñado para ser tolerante a fallos. Gestiona el flujo de datos de manera fiable, con mecanismos de reintento y manejo de errores que garantizan que los datos no se pierdan incluso si un procesador o un sistema externo falla.

- Procedencia de Datos (Data Provenance)

  Una de las características más valiosas de NiFi es su capacidad para rastrear la procedencia de los datos a lo largo de todo el flujo. Esto permite ver el origen de cada pieza de datos, qué transformaciones se le aplicaron, y dónde se almacenó finalmente. Esta auditoría es crucial para garantizar la calidad y la trazabilidad de los datos.

- Monitoreo y Control en Tiempo Real

  Aunque el proceso sea batch, NiFi permite monitorear el estado del flujo en tiempo real. Se puede visualizar el rendimiento, la cantidad de datos que fluyen por cada conexión y si hay cuellos de botella, lo que facilita la optimización y el diagnóstico de problemas.

3. Manejo del Desafío de la Alta Latencia en un Proceso Batch con Apache NiFi

La alta latencia es una característica inherente al procesamiento batch y no puede ser eliminada por completo. Sin embargo, se puede mitigar su impacto y optimizar el proceso utilizando las siguientes estrategias lógicas con NiFi

- Paralelización de la Ingesta

  NiFi permite ejecutar múltiples instancias de un mismo procesador o distribuir la carga de trabajo en un clúster. Al leer datos de diferentes fuentes simultáneamente o al procesar diferentes archivos de un directorio en paralelo, se reduce el tiempo total de procesamiento.

- Ajuste de la Programación (Scheduling)

  La latencia se puede percibir como menor si se ajusta la programación del proceso batch. En lugar de ejecutar el flujo una vez al día, se puede programar para que se ejecute cada hora, lo que reduce el tiempo de espera para que los datos estén disponibles.

- Optimización del Flujo

  Utilizar procesadores de NiFi diseñados para la eficiencia, como aquellos que procesan registros en grupos o utilizar procesadores de Record que son más eficientes que los procesadores que operan a nivel de flujo de archivos.

- Uso de la Memoria y el Almacenamiento Intermedio

  NiFi utiliza un repositorio de contenido “Content Repository” que almacena temporalmente los datos. Esto permite que los procesadores trabajen de manera asíncrona y fluida, evitando que los cuellos de botella detengan todo el proceso.

4. Implementación de un Flujo de Datos en Apache NiFi para Consolidación de Información

Para consolidar datos dispersos de bases de datos, archivos locales y APIs en un solo data warehouse utilizando Apache NiFi, se seguiría una serie de pasos lógicos en el lienzo de NiFi:

Paso 1: Ingesta de Datos desde Múltiples Fuentes (Extracción)

Se utilizaría un procesador de NiFi para cada tipo de fuente:

- Archivos Locales: Se usaría el procesador GetFile para leer archivos de un directorio específico.
- Bases de Datos: Se emplearía el procesador QueryDatabaseTable o ExecuteSQL para ejecutar consultas SQL y extraer los datos de las tablas deseadas.
- APIs: Se usaría el procesador Invoke HTTP para realizar peticiones a los endpoints de la API y obtener los datos en formato JSON o XML.

Paso 2: Estandarización y Transformación (Transformación)

Una vez que los datos de cada fuente están en el flujo de NiFi, se deben unificar en un formato común (por ejemplo, Avro o JSON).

- Se usarían procesadores como ConvertRecord o JoltTransformJSON para convertir los datos de cada fuente al formato estándar deseado.
- Se podrían usar procesadores como “QueryRecord” para limpiar, filtrar o enriquecer los datos utilizando expresiones similares a SQL.

Paso 3: Consolidación y Enriquecimiento

Una vez estandarizados, los flujos de datos de las diferentes fuentes se dirigirán a un punto común.

- Se podría usar un procesador como “MergeContent” para combinar los flujos de datos en un solo lote más grande si fuera necesario, para mejorar el rendimiento de la carga.
- Se podrían usar procesadores como “UpdateAttribute” para agregar metadatos o identificadores comunes a todos los registros, lo que facilita la carga en el data warehouse.

Paso 4: Carga de Datos (Carga)

Finalmente, el flujo de datos consolidado se cargaría en el data warehouse.

- Se utilizará un procesador como “PutDatabaseRecord” para insertar los datos estandarizados en las tablas del data warehouse.
- Alternativamente, si el data warehouse está en la nube como por ejemplo, Google BigQuery, Amazon Redshift o Snowflake, se usarían los procesadores específicos del servicio como “PutBigQuery” para realizar la carga de manera eficiente.

El proceso lógico en el lienzo de NiFi sería una cadena visual de estos procesadores, donde la salida de uno se conecta a la entrada del siguiente, creando un flujo de datos completo y auditable.