Curso

Data Engineering

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Centro de e-Learning UTN.BA

Módulo 2:

Procesamiento avanzado y seguridad de datos

Unidad 2:

Procesamiento en tiempo real

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Presentación

En esta unidad, exploraremos el campo del procesamiento de datos en tiempo real. Comenzaremos comprendiendo los fundamentos de este tipo de procesamiento y su importancia en comparación con los enfoques de procesamiento por lotes. También analizaremos los beneficios y desafíos asociados con el procesamiento en tiempo real y detallaremos los componentes para implementar estas soluciones.

Por último, abordaremos conceptos y términos claves para comprender cómo se implementa el procesamiento de datos en tiempo real de forma eficiente.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bloques temáticos

1. Descripción
2. Componentes
3. Conceptos

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Descripción

El procesamiento de datos en tiempo real, o en streaming, es una disciplina fundamental en la ingeniería de datos, que se enfoca en el manejo eficiente y rápido de flujos continuos de datos para generar informes o respuestas automáticas en tiempo real o casi en tiempo real. En este caso, se debe realizar el procesamiento sobre los datos a medida que se generan o se reciben, en lugar de procesarlos de manera batch (por lotes) en un momento posterior. Esta forma de procesamiento es esencial para aplicaciones que requieren respuestas rápidas y acciones en tiempo real, como sistemas de detección de fraudes, monitoreo de redes, análisis de sensores y muchas otras aplicaciones en tiempo real.

En el procesamiento en streaming, **cada nuevo dato se procesa a medida que llega**. A diferencia del procesamiento batch, no hay que esperar hasta el siguiente intervalo de procesamiento, sino que los datos se procesan como unidades individuales en tiempo real, en lugar de por lotes.

El procesamiento en streaming es ideal para operaciones en las que el tiempo es un factor crítico y que requieren una **respuesta instantánea** en tiempo real. Por ejemplo, un sistema que vigila un edificio en busca de humo y calor necesita activar alarmas y desbloquear puertas para que los residentes puedan escapar inmediatamente en caso de incendio. Otro ejemplo puede ser una solución de control del tráfico en tiempo real, que consume datos de sensores para detectar volúmenes de tráfico elevados. Estos datos podrían utilizarse para actualizar dinámicamente un mapa que muestre la congestión, o iniciar automáticamente carriles de alta ocupación u otros sistemas de gestión del tráfico.

A continuación, se presenta una tabla de comparación entre el procesamiento batch y en streaming

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

	Batch	Tiempo real
Alcance	Procesa todos los datos de un dataset	Procesa el dato más reciente, o aquellos que están dentro de una ventana de tiempo móvil (por ej: de los últimos 30 segundos)
Tamaño	Es ideal para grandes volúmenes de datos	Es adecuado para registros individuales o “micro lotes” de datos
Performance	El procesamiento puede demorar horas o minutos	El procesamiento demorar segundos o milisegundos

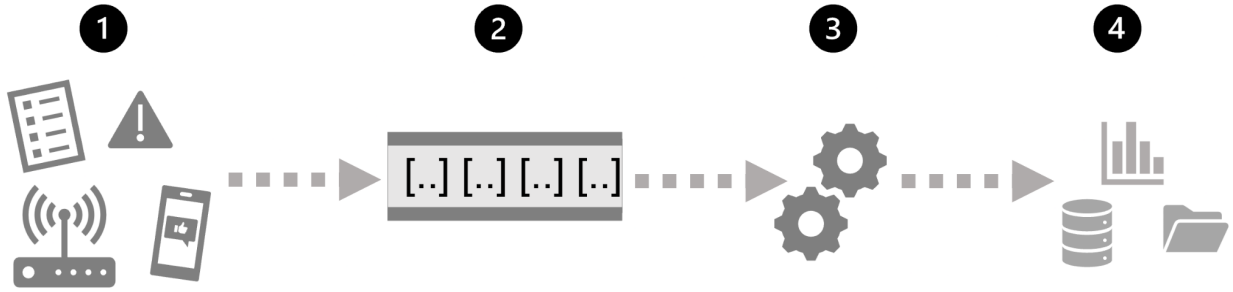
Componentes

En este caso, este tipo de procesamiento consume eventos o mensajes de sistemas como una cola de mensajería (message queue), un repositorio de archivos, bases de datos, etc.

Por lo general, estos datos entrantes suelen llegar en un formato no estructurado o semiestructurado, como JSON. Luego los datos son procesados, aplicando filtros, transformaciones, agregaciones, etc. Por último, los resultados son dirigidos hacia algún sistema como un almacén de datos analíticos, optimizado para el análisis y la visualización, que soporte altos volúmenes de escritura.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

En el procesamiento en tiempo real, nos vamos a encontrar con los siguientes componentes:



1. Fuente de datos, que genere o emita eventos o mensajes de forma continua.
2. Ingesta de mensajes en tiempo real. Se trata de un componente para capturar y almacenar mensajes en tiempo real que serán consumidos por otros procesos de forma inmediata. En casos sencillos, podría implementarse como un repositorio en el que los mensajes nuevos se depositan en una carpeta o directorio. Pero a menudo, se requiere algo más robusto como una cola o un broker de mensajes, que actúe como un búfer.

   Una cola o un broker encapsula la lógica para garantizar que los datos de los eventos sean procesados de forma ordenada y que cada evento se procese una sola vez. Algunos ejemplos son tecnologías como Apache Kafka, Apache Pulsar, MQTT, RabbitMQ.
3. Motor de procesamiento: Tras capturar los mensajes en tiempo real, el motor se encarga de procesarlos, aplicando filtros, agregaciones y preparando los datos para el análisis. El motor puede operar sobre registros individuales, o bien sobre “micro-lotes” de datos a partir de ventanas temporales (por ej. eventos de los últimos 30 segundos).
4. Un sistema de destino para depositar los resultados, puede ser en un repositorio de archivo, una base de datos SQL o NoSQL, otra cola o broker para continuar con el procesamiento, etc.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Conceptos

El procesamiento en tiempo real se basa en una serie de conceptos y términos fundamentales que sustentan su funcionamiento eficiente y ágil. Examinaremos algunos de ellos:

Tópico

Un **tópico** actúa como un **canal** de comunicación que **recibe eventos** o mensajes y los **distribuye** a las aplicaciones o sistemas interesados en ellos. Es una forma de organizar y categorizar los datos en un sistema de procesamiento en tiempo real.

Cuando un evento ocurre y necesita ser transmitido a otros componentes del sistema, se publica en un tópico específico. Los eventos pueden incluir información como transacciones, actualizaciones de estado, alertas, cambios de datos, entre otros. Cada evento está asociado a un tópico que define su naturaleza o contexto.

Un tópico permite que múltiples aplicaciones o sistemas se **suscriban de manera asincrónica** a él. Esto significa que los consumers pueden recibir y procesar los eventos a su propio ritmo, sin necesidad de estar sincronizados entre sí. Cada consumers interesado en un tópico en particular puede suscribirse a él y recibir los eventos que se publiquen allí.

Por lo general, los mensajes dentro de un tópico se ordenan siguiendo el principio **FIFO** (First-In, First-Out). Cuando un evento es publicado en un tópico, se agrega al final de la cola de eventos pendientes en ese tópico. Los consumers que están suscritos al tópico recibirán los eventos en el orden en que fueron publicados, asegurando así la integridad de la secuencia de eventos.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Sin embargo, es importante tener en cuenta que en algunos sistemas de procesamiento en tiempo real más avanzados, existe la posibilidad de modificar el orden de procesamiento de eventos según ciertas reglas o prioridades definidas.

Particiones

Para un manejo eficiente de altos volúmenes de datos y generados a grandes velocidades, es necesario trabajar con plataformas distribuidas.

En tecnologías como Apache Kafka, un tópico internamente se divide en particiones. En vez de tener un solo canal, vamos a contar con varios canales para el mismo tópico. Las particiones permiten distribuir y procesar la carga de trabajo de manera eficiente y paralela, lo que facilita la escalabilidad y el rendimiento en sistemas de alto volumen y alta velocidad.

Los eventos en cada partición se ordenan según el principio FIFO, como ya se explicó hace instantes, y pueden ser procesados de acuerdo al tiempo en el que arribaron.

Los mensajes pueden arribar a una partición en particular de forma aleatoria, o la aplicación que publique los datos debe definir de antemano a qué partición enviará cada mensaje.

Existe la posibilidad de que cada partición haga referencia a una categoría específica dentro del mismo tópico. Cada partición se ocupa de un conjunto específico de eventos relacionados con la categoría correspondiente. Esto permite que los consumers se suscriban a las particiones que les interesen y reciban sólo los eventos relevantes para esa categoría. Por ejemplo, si tenemos un tópico sobre "Viajes en Uber", dentro del mismo, podemos tener particiones que representen diferentes ubicaciones geográficas, como "Nueva York", "Los Ángeles" y "Londres".

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Patrón Pub/Sub

El patrón pub/sub (publicador/suscriptor) es un modelo de comunicación utilizado en sistemas distribuidos y en el procesamiento en tiempo real. Este patrón facilita la transmisión eficiente de eventos o mensajes desde los **“producers”** a los **“consumers”**, **“suscriptores”**.

En el patrón pub/sub, los producers son responsables de publicar los eventos en tópicos específicos. Cada evento se envía al tópico correspondiente sin necesidad de conocer quiénes son los consumers. Los producers no están directamente vinculados a los consumers, lo que brinda una mayor flexibilidad y escalabilidad en el sistema.

Por otro lado, los consumers se suscriben a los tópicos que les interesan. Al suscribirse, indican su interés en recibir los eventos relacionados con esos tópicos. Cuando un evento se publica en un tópico, todos los consumers suscritos a ese tópico recibirán el evento y podrán procesarlo de acuerdo a sus necesidades.

Este modelo de comunicación asincrónica permite que los producers y consumers operen de manera independiente y sin acoplamiento directo. Los producers pueden continuar publicando eventos sin preocuparse por quién los recibirá, mientras que los consumers pueden recibir eventos de múltiples productores sin necesidad de saber de dónde provienen.

Producer y consumer

Se tratan de aplicaciones capaces de enviar y recibir datos respectivamente. Se pueden implementar utilizando alguna librería de un lenguaje de programación e

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

incluso sistemas o plataformas que abstraen al desarrollador en el uso de librerías y bibliotecas.

Flujo de eventos

Un stream o flujo de eventos se refiere a una secuencia continua de eventos o mensajes que se generan y transmiten de manera constante, y en tiempo real. Estos eventos pueden representar diferentes tipos de información, como registros de actividad, transacciones, lecturas de sensores, clics de usuarios, tweets, y mucho más.

Los eventos se generan y transmiten de manera continua, sin interrupciones. El flujo puede ser constante o variar en su ritmo, pero siempre está en movimiento.

Comparado al procesamiento batch, donde se opera sobre un conjunto de datos limitado por intervalos de tiempo, en los stream de eventos nos encontramos con dataset sin límites, que no tiene fin.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Conclusión

En esta última unidad, hemos explorado el mundo del procesamiento de datos en tiempo real y hemos adquirido una comprensión sólida de los conceptos fundamentales y sus componentes clave.

Hemos abarcado desde la ingesta y recepción de mensajes pasando por su procesamiento y su almacenamiento o redirección hacia otros sistemas.

El procesamiento en tiempo real no se puede considerar una mejora o un reemplazo al procesamiento batch. Es importante saber que cada uno tiene un alcance y un propósito específico. Existen soluciones que combinan ambos tipos de procesamientos o solo usan una de ellas de acuerdo al caso de uso.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bibliografía utilizada y sugerida

- Explore fundamentals of real-time analytics - Training. (s. f.). Microsoft Learn. https://learn.microsoft.com/en-us/training/modules/explore-fundamentals-stream-processing/ Kleppmann, M. (s. f.).
- Stream processing [Ebook]. En Designing Data-Intensive Applications. O'Reilly. https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/
- Real-time processing - Azure Architecture Center. (s. f.). Microsoft Learn. https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/real-time-processing

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning