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 1:

Procesamiento de datos

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

Procesamiento de datos puede ser un término amplio. Considerando que en las unidades anteriores nos enfocamos en la extracción de fuentes de datos y en su almacenamiento en un sistema centralizado, el procesamiento implica la obtención de información a partir de los datos crudos.

En primer lugar, hay que aplicar tareas “tradicionales” como tratar valores nulos, ya sea eliminándolos o reemplazándolos con otro valor, por ejemplo. Otra tarea tradicional puede ser la eliminación de registros duplicados. Las tareas en esta primera instancia implica obtener datos de calidad, aptos para continuar con tareas de procesamiento y analítica avanzada.

La siguiente etapa de procesamiento está vinculada a la organización o al negocio. Una vez que contemos con datos de calidad, el siguiente paso es enriquecer y obtener información valiosa para la organización. Aquí se pueden crear nuevas columnas que resulten de aplicar cálculos específicos, cruzar diferentes datos de diferentes fuentes, etc.

Todo esto lo veremos en un material aparte con un enfoque práctico.

Para el procesamiento de datos, hay que tener en cuenta algunos conceptos que iremos viendo a lo largo de esta unidad teórica, como:

- el procesamiento distribuido para el caso de grandes volúmenes de datos
- el procesamiento batch y en streaming, cada uno ofrece tiempos de respuesta diferentes y operan sobre un lote de datos o sobre registros individuales, respectivamente

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

- data pipelines y orquestación, por último, para asegurar que las tareas del proceso se ejecutan de forma ordenada y con robustez.

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. Big data y procesamiento distribuido
2. Tipos de procesamiento
3. Pipelines de datos y orquestación

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

Big data y procesamiento distribuido

En la unidad anterior, hablamos sobre Big Data con foco en el almacenamiento distribuido, por medio de clústers como infraestructura o plataforma. Un clúster nos permite procesar grandes volúmenes de datos de forma paralela y distribuida.

En vez de alojar y procesar datos en una sola computadora, es posible hacerlo por medio de un conjunto de computadores (clúster) para acelerar la ejecución. Esto es posible gracias a un reconocido paradigma de big data, MapReduce.

MapReduce es un paradigma de programación para procesar y analizar grandes volúmenes de datos de manera eficiente.

La idea detrás de MapReduce es dividir una tarea de procesamiento en dos **etapas principales: Map y Reduce.** En la etapa de Map, los datos se dividen en pequeñas partes y se aplican una serie de transformaciones o cálculos a cada una de ellas de manera independiente. Estas transformaciones generan pares clave-valor como resultado.

Una vez completada la etapa de Map, los pares clave-valor se agrupan y se envían a la etapa de Reduce. En esta etapa, los datos se procesan y se combinan para obtener un resultado final. La fase de Reduce puede implicar la suma, el promedio, el conteo u otras operaciones que permiten obtener información resumida de los datos.

La ventaja clave de MapReduce es su capacidad para procesar datos de manera distribuida, lo que significa que puede aprovechar el poder de procesamiento de múltiples computadoras o servidores en paralelo. Esto permite manejar grandes volúmenes de datos de manera más eficiente y reducir el tiempo de procesamiento.

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

Este paradigma se dió a conocer cerca del 2004. Era posible implementarlo por medio de tecnologías como Hadoop MapReduce y Java.

Con el paso del tiempo fueron surgiendo tecnologías cada vez más potentes. Actualmente, una de las tecnologías más populares y potentes en Big Data y Data Engineering es Apache Spark.

Apache Spark es un **motor analítico unificado** para el procesamiento de datos a gran escala. Es posible utilizarlo por medio de lenguajes como Java, Scala, Python y R. Se trata de un motor unificado porque ofrece un conjunto de herramientas de alto nivel, como Spark SQL para el procesamiento de datos estructurados, MLlib para machine learning, GraphX para el procesamiento de grafos y Structured Streaming para el procesamiento en tiempo real.

Claramente, Spark opera de forma distribuida sobre un clúster de servidores. Divide el trabajo en múltiples nodos de un clúster, lo que permite procesar los datos de forma paralela y acelerar el tiempo de ejecución.

Una de las características más destacadas de Spark es su capacidad para mantener los datos **en memoria**, lo que significa que puede acceder a ellos rápidamente para su procesamiento. Esto es especialmente útil cuando necesitas realizar múltiples operaciones en los mismos datos, ya que evita tener que leerlos repetidamente desde el disco, lo que ralentiza el proceso.

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

Tipos de procesamiento

Procesamiento batch

El procesamiento de datos batch, o por lotes, opera sobre un **volúmen de datos definido,** es decir sobre un lote de datos. Su ejecución se hace en **intervalos programados**, por ejemplo cada hora, cada seis horas, cada día, etc. Además la ejecución de estos procesos puede durar minutos u horas.

En este caso, los datos se recopilan y se almacenan hasta que se haya reunido una cantidad suficiente para procesarlos de una sola vez. Por ejemplo, si tenemos que trabajar con registros de ventas de un supermercado almacenados en un sistema, en lugar de procesar cada registro de venta a medida que ocurre, el procesamiento batch recopila todos los registros en un lote, por ejemplo, cada hora o cada día, y luego los procesa juntos como un conjunto.

Es especialmente útil cuando no se necesita una respuesta en tiempo real y se pueden tolerar ciertos retrasos en el procesamiento. Es eficiente para trabajar con grandes volúmenes de datos, ya que se pueden aplicar optimizaciones y técnicas de procesamiento paralelo para acelerar el proceso.

Procesamiento en streaming

El procesamiento de datos en streaming se refiere a la forma en que los datos se procesan de **manera continua** y en tiempo real **a medida que se generan o se reciben.** A diferencia del procesamiento por lotes, donde los datos se recopilan y se procesan en conjuntos, el procesamiento en streaming opera sobre los datos de forma **inmediata**, a medida que fluyen.

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

El procesamiento en streaming implica recibir, procesar y analizar continuamente los datos a medida que llegan. Esto se logra mediante el uso de sistemas y herramientas específicas, como Apache Kafka, Apache Flink o Apache Spark Streaming, que están diseñadas para manejar flujos continuos de datos.

A medida que los datos de streaming ingresan al sistema, se aplican operaciones en tiempo real para filtrar, transformar, enriquecer o agregar información a medida que los eventos se procesan. Estas operaciones pueden incluir cálculos, correlaciones, detección de anomalías o cualquier otro tipo de procesamiento requerido para extraer información valiosa de los datos en tiempo real.

Es especialmente útil en situaciones donde se requiere una **baja latencia**, es decir, una **respuesta inmediata** a los eventos que ocurren. Puede ser utilizado en aplicaciones como análisis de fraudes en tiempo real, monitoreo de sistemas en tiempo real, etc.

Pipelines y orquestación

Las “data pipelines” generalmente consisten en varias **tareas o acciones** que deben ejecutarse de **forma ordenada** para lograr el resultado deseado, **mover datos de un lugar a otro**. La salida de cada tarea o acción suele ser la entrada de la tarea siguiente.

Por lo general, las tareas deben ejecutarse en un orden específico. Puede haber casos en los que cada paso se ejecuta y finaliza antes de que comience el siguiente, asegurando un flujo secuencial y ordenado. Sin embargo, también es posible utilizar enfoques de procesamiento paralelo o distribuido en un data pipeline, donde los pasos se ejecutan de manera concurrente. En este caso, los pasos pueden comenzar a ejecutarse tan pronto como haya suficiente cantidad

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

de datos disponibles para su procesamiento, sin tener que esperar a que el paso anterior haya finalizado completamente.

El proceso de construcción de un data pipeline implica varios pasos:

- Extracción de las fuentes de datos
- Transformación de los datos, donde se aplican reglas y técnicas para filtrar, limpiar, enriquecer los datos
- Almacenamiento de los datos, donde se cargan y persisten los datos ya procesados para que puedan ser consumidos por los interesados.
- Orquestación: Un data pipeline también puede involucrar la orquestación de diferentes procesos y tareas en un flujo de trabajo secuencial. Esto implica programar y coordinar la ejecución de los diferentes pasos del pipeline, asegurándose de que se realicen en el orden correcto y que los datos fluyan de manera eficiente y confiable.

Es importante tener en cuenta varios aspectos clave para garantizar su eficiencia, confiabilidad y escalabilidad de los data pipelines.

En cuanto a la **gestión de errores y tolerancia a fallos**, los data pipelines pueden enfrentar diferentes tipos de errores, como fallas en la extracción de datos, errores de transformación o problemas de conectividad. Es fundamental implementar mecanismos de gestión de errores y tolerancia a fallos, como **mecanismos de reintentos**, registros o **logs** detallados de errores y alertas. Estas medidas ayudan a garantizar la **integridad y confiabilidad** del pipeline, minimizando la pérdida de datos y el impacto en los procesos de análisis posteriores.

Respecto al **monitoreo** y métricas, un data pipeline debe ser monitoreado de cerca para detectar posibles problemas o cuellos de botella. Es importante establecer **métricas y alertas** para supervisar el **rendimiento**, la **latencia**, el

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

volumen de datos procesados y otros indicadores relevantes. Esto permite identificar y solucionar problemas de manera proactiva, asegurando que el pipeline funcione de manera óptima.

El último aspecto a destacar es la **modularidad y reutilización**. Es recomendable diseñar el data pipeline de manera modular y reutilizable. Esto implica dividir el pipeline en componentes más pequeños y funcionales, lo que facilita su mantenimiento, escalabilidad y reutilización en otros proyectos. El diseño modular también permite realizar pruebas y depuración más efectivas, ya que cada componente puede ser evaluado de forma individual.

Ya hemos visto términos como ETL, ELT, procesamiento batch y en streaming. Cada uno de estos es considerado un tipo de data pipeline.

Las plataformas de orquestación de flujos de trabajo como Apache Airflow, Prefect, entre otros, ofrecen capacidades muy útiles para programar, monitorear y ejecutar tareas en un pipeline de datos. Facilitan la definición y el control de tareas complejas, permiten la integración con diferentes sistemas y tecnologías, y proporcionan capacidades de monitoreo y administración para garantizar la confiabilidad y el rendimiento del pipeline.

Los data pipelines constituyen una parte fundamental en el flujo de trabajo de la ingeniería de datos y desempeñan un papel crucial en la obtención de información valiosa a partir de los datos.

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

Conclusión

El procesamiento distribuido ofrece grandes ventajas al momento de trabajar con grandes volúmenes de datos. En varias ocasiones, es conveniente trabajar con un clúster de servidores con capacidades de cómputo comunes en vez de utilizar una gran computadora con alto nivel de recursos.

Como Data Engineers vamos a desarrollar data pipelines para tomar datos de un lugar y depositarlos en otro de acuerdo a las necesidades de la organización. Ese data pipeline puede tratarse de un proceso ETL para cargar un data warehouse, un proceso ELT para almacenar y transformar datos en un data lake, un procesamiento de eventos en tiempo real, etc. Ese data pipeline debe ser lo suficientemente robusto para manejar volúmenes de datos variables, garantizar la integridad de los datos y minimizar la pérdida de los mismos. Es importante elegir las herramientas y tecnologías adecuadas para el proyecto y asegurarse de que el pipeline sea modular, escalable, reutilizable y fácil de mantener. Además, es fundamental monitorear y medir su rendimiento y confiabilidad para garantizar que funcione de manera eficiente y efectiva.

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

- Apache SparkTM - Unified Engine for large-scale data analytics. (s. f.). https://spark.apache.org/
- Batch processing - Azure Architecture Center. (s. f.). Microsoft Learn. https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/batch-processing
- Dean, J. (2004). MapReduce: Simplified Data Processing on Large Clusters. Google Research. https://research.google/pubs/pub62/
- 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
- What is MapReduce? | IBM. (s. f.). IBM. https://www.ibm.com/topics/mapreduce

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