MILTON GOYA

MACHINE LEARNING ENGINEERING

BIG DATA PIPELINES

AULA 04

SUMÁRIO

O QUE VEM POR AÍ?

Você sabe qual é a diferença entre batch pipeline e stream pipeline no mundo dos big data pipelines? Os pipelines de dados em Big Data assumem um papel crucial na transformação de dados brutos em insights valiosos. Enquanto os pipelines batch processam dados em lotes, ideais para análises retrospectivas, os pipelines de stream lidam com dados em tempo real, permitindo insights instantâneos e ações imediatas. Conhecer as diferenças entre esses dois tipos de pipelines é essencial para uma estratégia eficaz de processamento de dados em Big Data.

HANDS ON

Analise o benchmarking do Yahoo! e busque responder às seguintes perguntas:

- Quais foram os principais fatores que impactaram a performance do Apache Spark nos benchmarks da Yahoo e como eles se comparavam aos do Apache Flink?
- Como o Apache Flink se comportou em termos de latência e throughput em comparação com o Apache Storm, especialmente quando o "acking" estava desativado?
- Quais aspectos do processamento de eventos em tempo real foram destacados como pontos fortes do Flink nos benchmarks realizados?
- Existem recomendações específicas de configuração do Apache Flink para melhorar o desempenho em operações de janelas com base nos estudos de benchmark?
- Em quais cenários específicos o Apache Spark mostrou ser mais eficiente que o Apache Flink, segundo os benchmarks?
- Quais foram as principais diferenças observadas entre Spark e Flink no que diz respeito à escalabilidade e ao consumo de recursos em grandes conjuntos de dados?
- Quais são as implicações práticas das diferenças de desempenho entre Spark e Flink para desenvolvedores(as) e administradores(as) de sistemas que estão escolhendo uma plataforma para processamento de dados em larga escala?

SAIBA MAIS

Stream vs Batch

Nos últimos anos, observamos um crescimento explosivo na quantidade de dados gerados diariamente. Esses dados são provenientes de diversas fontes, como softwares, câmeras, sensores, rastreadores de atividade e satélites, entre outros, gerando dados em formatos variados. Para processar e gerenciar essa vasta quantidade de informações, são necessários pipelines de dados robustos e eficientes.

Assim, as arquiteturas de Pipelines de Dados são essenciais para entender como organizar, entregar e gerenciar o fluxo de dados. Existem dois métodos principais de movimentação de dados: processamento em lote (batch processing) e processamento em fluxo contínuo (stream processing).

No processamento em lote (batch processing), os dados são agrupados em lotes que são migrados da fonte para o destino em intervalos regulares ou em uma única execução. Este método é eficiente para tarefas que não requerem dados em tempo real e podem ser agendadas para horários de menor demanda do sistema.

Por outro lado, o processamento em fluxo contínuo (stream processing) permite a movimentação de dados em tempo real. Ele coleta dados continuamente de diversas fontes, como eventos de sensores, sistemas de mensagens ou fluxos de alterações de um banco de dados. Este método é ideal para aplicações que necessitam de dados atualizados instantaneamente, como monitoramento em tempo real e análises instantâneas.

Essa distinção entre processamento em lote ou fluxo contínuo é crucial, pois permite a escolha da arquitetura de pipeline de dados mais adequada para cada necessidade específica de processamento e análise de grandes volumes de dados.

Definição e Importância dos Big Data Pipelines

Os pipelines de Big Data são essenciais para transformar e movimentar grandes volumes de dados de diversas fontes para destinos finais, como Data Lakes, Data Warehouses e bancos de dados relacionais. Eles podem lidar com dados semi-estruturados, estruturados e não estruturados, tornando-se uma ferramenta versátil para diferentes necessidades de processamento de dados.

O Que é um Pipeline de Dados?

Um Pipeline de Dados pode ser descrito como uma sequência de etapas de processamento de dados. Se os dados não estiverem carregados atualmente na plataforma, eles devem ser ingeridos no início do pipeline. Isso é seguido por uma série de etapas, cada uma fornecendo uma saída que, por sua vez, serve como entrada para a próxima etapa. Esse processo continua até que o pipeline esteja completo. Em alguns casos, é possível executar etapas independentes de forma concorrente.

Os Pipelines de Dados contêm três elementos-chave: uma fonte, uma etapa ou um conjunto de etapas de processamento e um destino (também conhecido como sink). Esses pipelines permitem o fluxo de dados de uma aplicação para um Data Warehouse, por exemplo, de um Data Lake para um sistema de processamento de pagamentos ou um banco de dados analítico.

Os Pipelines de Dados também podem ter a mesma fonte e destino, sendo o pipeline puramente sobre a modificação do conjunto de dados. Sempre que os dados são processados entre os pontos A e B (ou pontos C, B e D), um Pipeline de Dados faz a ligação entre esses pontos.

À medida que as organizações procuram construir aplicações com bases de código pequenas que atendam a um propósito específico, elas estão movendo dados entre um número crescente de aplicações, tornando a eficiência dos Pipelines de Dados uma consideração crítica em seu desenvolvimento e planejamento.

Figura 1 – Diagrama Simples de um Pipeline de Dados
Fonte: Elaborado pelo autor (2024)

A figura 1 ilustra um pipeline de dados simples, demonstrando o fluxo de dados desde a fonte até o destino através de duas etapas de processamento intermediárias. Esse fluxo pode ser descrito como segue:

- **Fonte**: o ponto de entrada dos dados no pipeline.

- **Etapa de Processamento 1**: a primeira etapa em que os dados são processados.
- **Etapa de Processamento 2**: a segunda etapa de processamento dos dados.
- **Destino**: o ponto final em que os dados processados são armazenados ou utilizados.

Esse pipeline mostra como os dados se movem através de uma sequência de etapas de processamento, cada uma entregando uma saída que serve como entrada para a próxima etapa, até atingir o destino final.

O que é um Pipeline de Big Data?

Nos últimos anos, a variedade, o volume e a velocidade dos dados cresceram consideravelmente. Desenvolvedores(as) e arquitetos(as) de sistemas precisaram se adaptar ao conceito de Big Data. Assim, de forma simplificada, Big Data refere-se ao grande volume de dados que deve ser gerenciado. Esse volume massivo cria oportunidades para casos de uso como relatórios em tempo real, alertas e análises preditivas, entre outros exemplos.

A diferença mais significativa entre pipelines de dados comuns e pipelines de Big Data é a flexibilidade para transformar grandes volumes de dados. Um pipeline de Big Data pode processar dados em fluxos (streams), lotes (batches) ou outros métodos, cada um com suas vantagens e desvantagens. Independentemente do método, um pipeline de dados precisa ser capaz de escalar conforme as necessidades da organização para ser eficaz como um pipeline de Big Data. Sem escalabilidade, o sistema pode levar dias ou semanas para concluir suas tarefas.

Características Principais de um Pipeline de Big Data

Algumas características essenciais que fazem um Pipeline de Big Data se destacar são:

- **Arquitetura Escalável Baseada em Nuvem**: os Pipelines de Big Data dependem da Nuvem para permitir que usuários escalem automaticamente recursos de armazenamento e computação, para cima ou para baixo. Enquanto os pipelines tradicionais não são projetados para lidar com

múltiplas cargas de trabalho simultaneamente, os Pipelines de Big Data possuem uma arquitetura em que os recursos computacionais são distribuídos entre clusters independentes.

Esses clusters podem crescer em tamanho e número rapidamente, mantendo o acesso ao conjunto de dados compartilhado. Assim, é mais fácil prever o tempo de processamento dos dados, pois novos recursos podem ser adicionados instantaneamente para suportar picos no volume de dados. Os Pipelines de Dados baseados na Nuvem são elásticos e ágeis, permitindo que as empresas aproveitem diversas tendências.

Por exemplo: uma empresa que espera um aumento nas vendas durante o verão pode adicionar mais poder de processamento conforme necessário, sem precisar se planejar semanas antes para esse cenário. Na ausência de Pipelines de Dados elásticos, as empresas podem ter dificuldade em se adaptar rapidamente às tendências.

- **Arquitetura Tolerante a Falhas**: a falha em um Pipeline de Dados é uma possibilidade real enquanto os dados estão em movimento. Para mitigar os impactos em processos críticos, os Pipelines de Dados atuais oferecem um alto grau de disponibilidade e confiabilidade.

  Os Pipelines de Big Data são projetados com uma arquitetura distribuída que alerta usuários e fornece failover imediato em caso de falha de aplicação, falha de nó ou falha de outros serviços específicos. Se um nó falhar, outro nó dentro do cluster pode assumir imediatamente, sem a necessidade de grandes intervenções.

- **Transformação de Grandes Volumes de Dados**: como dados semiestruturados e não estruturados compõem cerca de 80% dos dados coletados pelas empresas, os Pipelines de Big Data devem ser equipados para processar grandes volumes de dados não estruturados (incluindo dados de sensores, arquivos de log e dados meteorológicos) e dados semiestruturados (como arquivos HTML, JSON e XML).

  Os Pipelines de Big Data podem ter que migrar e unificar dados de sensores, aplicativos, arquivos de log ou bancos de dados.

Frequentemente, os dados precisam ser padronizados, enriquecidos, filtrados, agregados e limpos – tudo praticamente em tempo real.

- **Análises e Processamento em Tempo Real**: os Pipelines de Big Data devem transformar, ingerir e analisar dados quase em tempo real para que as empresas possam encontrar e agir rapidamente com base em insights. Para começar, os dados precisam ser ingeridos sem demora de fontes como dispositivos IoT, bancos de dados, sistemas de mensagens e arquivos de log.

  Para bancos de dados, a captura de dados de alteração baseada em log (CDC) serve como o padrão ouro para produzir um fluxo de dados em tempo real. Os Pipelines de Dados em tempo real fornecem aos tomadores de decisão os dados mais recentes à sua disposição.

- **Gestão Autônoma**: os Pipelines de Big Data são construídos usando ferramentas que estão interligadas. De Data Warehouses e plataformas de integração de dados a Data Lakes e linguagens de programação, as equipes podem utilizar várias ferramentas para manter e desenvolver Pipelines de Dados de maneira autônoma e automatizada. Os Pipelines de Dados tradicionais geralmente exigem muito esforço e tempo para integrar um vasto conjunto de ferramentas externas para transferência, extração e análise de dados.

  A manutenção contínua pode ser demorada e causar gargalos que introduzem novas complexidades. Os Pipelines de Big Data também democratizam o acesso aos dados. Lidar com todos os tipos de dados é mais automatizado e fácil do que antes, permitindo que as empresas aproveitem os dados com menos pessoal interno e esforço.

- **Desenvolvimento de Pipelines de Dados Otimizado**: os Pipelines de Big Data são desenvolvidos seguindo os princípios de DataOps, uma metodologia que reúne vários processos e tecnologias para encurtar os ciclos de desenvolvimento e entrega.

  Como o DataOps lida com a automação dos Pipelines de Dados ao longo de seu ciclo de vida inteiro, os pipelines podem entregar dados pontualmente ao stakeholder correto. Ao alinhar a implantação e o

desenvolvimento do pipeline, você facilita a escalabilidade ou a alteração dos pipelines para incluir novas fontes de dados.

- **Processamento com Precisão**: a duplicação de dados e a perda de dados são alguns dos problemas comuns enfrentados pelos Pipelines de Dados. Os Pipelines de Big Data oferecem habilidades avançadas de checkpointing que garantem que nenhum evento seja processado duas vezes ou perdido. O checkpointing rastreia os eventos processados e como eles percorrem diferentes Pipelines de Dados.

  O checkpointing se integra com a funcionalidade de repetição de dados fornecida por várias fontes, permitindo que você volte ao ponto correto se ocorrer uma falha. Para fontes sem funcionalidade de repetição de dados, os Pipelines de Dados com mensagens persistentes podem fazer checkpoint e repetir os dados para garantir que foram processados apenas uma vez.

Tipos de Pipelines de Big Data

Existem diferentes tipos de Pipelines de Big Data comumente utilizados no mercado:

ETL (Extract, Transform, Load)

ETL é a arquitetura de pipeline de dados mais comum, sendo o padrão há várias décadas. Este processo envolve a extração de dados brutos de várias fontes, sua transformação em um formato predefinido e o carregamento para o destino – tipicamente um Data Mart ou um Data Warehouse corporativo.

Casos típicos de uso para Pipelines ETL incluem:

- Coletar grandes volumes de dados de diversas fontes internas e externas para oferecer uma visão holística das operações empresariais.
- Extrair dados de usuários de vários pontos de contato para consolidar todas as informações em um único lugar (geralmente em um sistema CRM).
- Vincular conjuntos de dados distintos para permitir análises mais profundas.

No entanto, a principal desvantagem da arquitetura ETL é a necessidade de reconstruir o pipeline de dados sempre que as regras de negócios são modificadas.

Para lidar com esse problema, outra abordagem ganhou destaque ao longo dos anos – a ELT.

ELT (Extract, Load, Transform)

ELT difere do ETL na sequência das etapas; o carregamento ocorre antes da transformação. Devido a essa pequena alteração, em vez de modificar grandes quantidades de dados brutos inicialmente, você os move diretamente para um Data Lake ou Data Warehouse. Isso permite que você estruture e processe os dados conforme necessário – a qualquer momento, parcial ou totalmente, inúmeras vezes ou apenas uma vez.

Casos de uso para a arquitetura ELT incluem:

- Cenários em que a velocidade de ingestão de dados é crucial.
- Situações em que não se tem certeza sobre o que será feito com os dados e como transformá-los.
- Cenários que envolvem grandes volumes de dados.

Pipeline de Dados em Lote (Batch)

O desenvolvimento do processamento em lote foi um passo crítico na construção de infraestruturas de dados confiáveis e escaláveis. Em 2004, o MapReduce, um algoritmo de processamento em lote, foi patenteado e subsequentemente integrado em sistemas de código aberto, como Hadoop, CouchDB e MongoDB (IBM, 2022).

Como o nome sugere, o processamento em lote carrega "lotes" de dados em um repositório durante intervalos de tempo definidos, que são tipicamente programados durante horários de menor movimento. Desta forma, outras cargas de trabalho não são impactadas, uma vez que os trabalhos de processamento em lote tendem a trabalhar com grandes volumes de dados, o que pode sobrecarregar o sistema geral. O processamento em lote é geralmente o pipeline de dados ideal quando não há uma necessidade imediata de analisar um conjunto de dados específico (como, por exemplo, contabilidade mensal) e está mais associado ao processo de integração de dados ETL, que significa "extrair, transformar e carregar" (IMB, 2022).

Os trabalhos de processamento em lote formam um fluxo de trabalho de comandos sequenciados, em que a saída de um comando se torna a entrada do próximo. Por exemplo: um comando pode iniciar a ingestão de dados, o próximo pode acionar a filtragem de colunas específicas e o subsequente pode lidar com a agregação. Essa série de comandos continuará até que a qualidade dos dados seja completamente transformada e reescrita em um repositório de dados (IBM, 2022).

No processamento em lote, você simplesmente coleta fragmentos de dados em um armazenamento temporário e os envia em grupo de acordo com uma programação. Você pode executar isso quando o acesso a esses dados não é urgente ou quando há problemas de latência intermitentes para lidar.

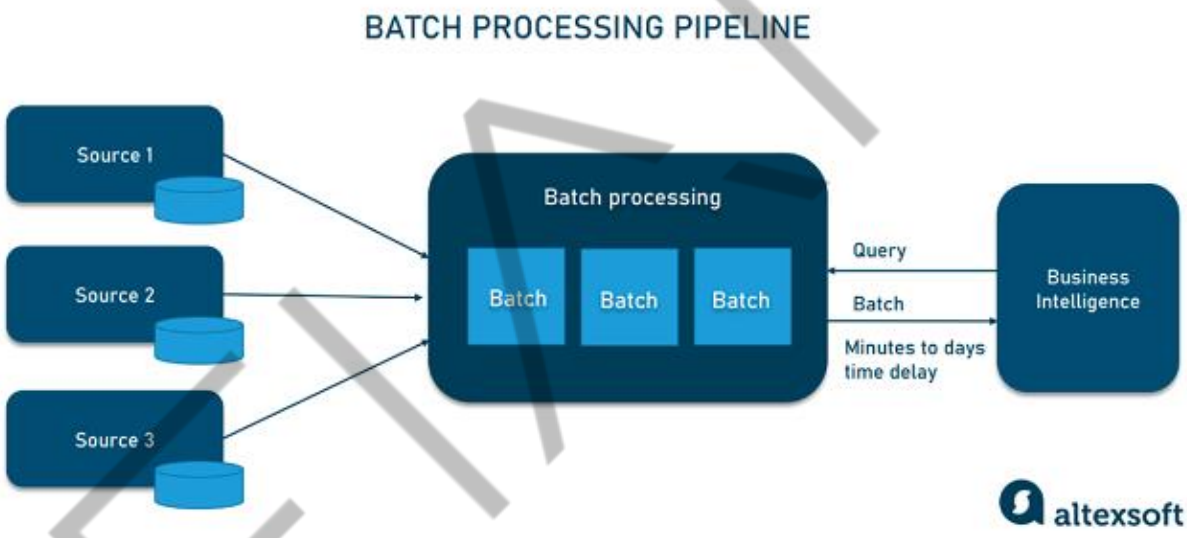

Figura 2 - Diagrama de um Pipeline Batch
Fonte: Altexsoft (2022)

Elementos do Pipeline

Sources (Fontes de Dados)

- **Source 1, Source 2, Source 3**: representam diferentes fontes de dados. Essas fontes podem ser bancos de dados, arquivos de log e sensores, entre outros. Os dados são coletados de várias origens e enviados para o processamento em lote.

Batch Processing (Processamento em Lote)

- **Batch**: o coração do pipeline, no qual o processamento em lote ocorre. Os dados são agrupados em "lotes" e processados em intervalos de tempo definidos. Cada lote é processado de forma sequencial ou paralela, dependendo da arquitetura e da necessidade do sistema.

Business Intelligence (Inteligência de Negócios)

- **Query (consulta)**: após o processamento dos lotes, os dados processados podem ser consultados (query) para análises e relatórios.
- **Batch (lote)**: os lotes processados são enviados para sistemas de inteligência de negócios (BI), onde são utilizados para gerar insights e tomar decisões informadas.

O fluxo de dados segue esta ordem:

Coleta de Dados

Os dados são coletados de múltiplas fontes (Source 1, Source 2, Source 3) e enviados para o módulo de processamento em lote.

Processamento em Lote

Dentro do módulo de processamento em lote, os dados são agrupados e processados em intervalos de tempo definidos. Isso pode envolver várias operações, como filtragem, agregação e transformação, entre outras.

Armazenamento e Consulta

Os dados processados em lotes são armazenados e podem ser consultados conforme necessário. A consulta (query) pode ser feita em tempo quase real, mas há um atraso natural (que pode variar de minutos a dias) devido ao processamento em lote.

Inteligência de Negócios

Os dados processados são integrados a sistemas de inteligência de negócios (BI) e utilizados para análises avançadas, relatórios e suporte à tomada de decisões estratégicas.

Pipeline de Dados de Streaming

Ao contrário do processamento em lote, os pipelines de dados de streaming – também conhecidos como arquiteturas orientadas a eventos – processam continuamente eventos gerados por várias fontes, como sensores ou interações de usuários dentro de um aplicativo. Os eventos são processados, analisados e então armazenados em bancos de dados ou enviados para análise posterior.

Utilizamos dados de streaming quando é necessário que os dados sejam continuamente atualizados. Por exemplo, aplicativos ou sistemas de ponto de venda precisam de dados em tempo real para atualizar o inventário e o histórico de vendas de seus produtos; dessa forma, vendedores(as) podem informar a consumidores(as) se um produto está em estoque ou não. Uma única ação, como a venda de um produto, é considerada um "evento" e eventos relacionados, como adicionar um item ao carrinho, são geralmente agrupados como um "tópico" ou "fluxo". Esses eventos são então transportados por sistemas de mensagens ou corretores de mensagens, como a oferta de código aberto, Apache Kafka (IBM, 2022).

Como os eventos de dados são processados pouco depois de ocorrerem, os sistemas de processamento de streaming têm menor latência do que os sistemas de processamento em lote, mas não são considerados tão confiáveis quanto os sistemas de processamento em lote, uma vez que as mensagens podem ser descartadas acidentalmente ou passar muito tempo na fila. Corretores de mensagens ajudam a resolver essa preocupação por meio de reconhecimentos, em que uma pessoa consumidora confirma o processamento da mensagem ao corretor para removê-la da fila (IBM, 2022).

O processamento em tempo real lida com dados que são movidos para processamento e armazenamento no momento em que são gerados, como um feed de dados ao vivo. O mecanismo de processamento de streaming pode fornecer saídas do pipeline de dados para armazenamentos de dados, CRMs e aplicativos de marketing. Do ponto de vista da implementação, o processamento de dados em streaming pode utilizar microlotes que podem ser concluídos em janelas de tempo curtas.

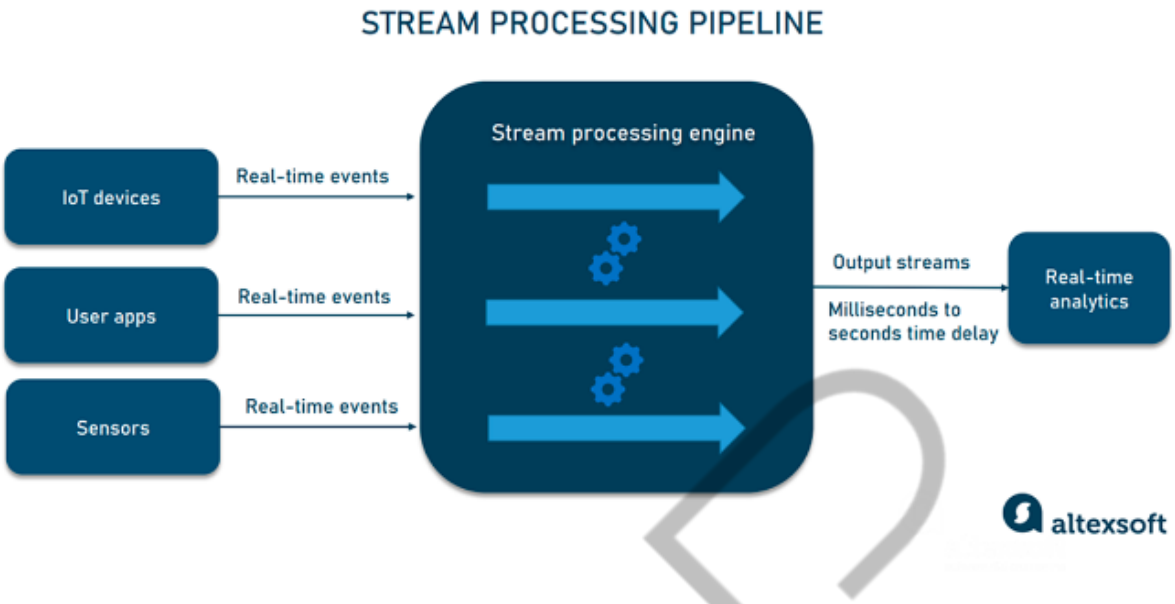

Figura 3 - Diagrama de um pipeline stream
Fonte: Altexsoft (2022)

Elementos do Pipeline

Fontes de Dados (Sources):

- **Dispositivos IoT**: esses dispositivos geram eventos em tempo real que são enviados para o pipeline.
- **Aplicativos de Usuário (User apps)**: aplicativos que interagem com usuários e geram eventos em tempo real, como cliques, ações e transações.
- **Sensores**: dispositivos que monitoram condições ambientais ou operacionais e enviam dados continuamente.

Eventos em Tempo Real (Real-time events):

- Os dados provenientes das fontes de dados são eventos gerados em tempo real. Esses eventos são enviados diretamente para o motor de processamento de fluxo.

Motor de Processamento de Fluxo (Stream processing engine):

- Este componente central do pipeline processa os eventos em tempo real conforme eles chegam. O processamento pode incluir operações como filtragem, agregação, transformação e análise dos dados.

- O motor de processamento de fluxo é projetado para lidar com grandes volumes de dados com baixa latência, garantindo que os dados sejam processados quase instantaneamente.

Saídas do Fluxo (Output streams):

- Após o processamento, os dados são enviados como fluxos de saída (output streams) para a próxima etapa do pipeline.

Análise em Tempo Real (Real-time analytics):

- Os fluxos de saída são enviados para sistemas de análise em tempo real, onde podem ser usados para gerar insights imediatos, relatórios e visualizações. Essa análise permite que as organizações tomem decisões rápidas e informadas com base nos dados mais recentes.

Latência (Milliseconds to seconds time delay):

- A latência do pipeline de processamento em fluxo é medida em milissegundos a segundos, indicando o tempo que leva desde a geração do evento até a conclusão do processamento e a entrega dos dados analisados.

O QUE VOCÊ VIU NESTA AULA?

Nessa aula, você aprendeu sobre as diferenças entre os pipelines de dados batch e stream no contexto de big data.

Compreendemos que os pipelines batch processam dados em lotes, ideais para análises retrospectivas e não urgentes, enquanto os pipelines stream lidam com dados em tempo real, permitindo insights instantâneos e ações imediatas.

Por fim, também exploramos como escolher a arquitetura de pipeline adequada depende das necessidades específicas de processamento e análise de grandes volumes de dados.

REFERÊNCIAS

ALTEXSOFT. **Data Pipeline**: Components, Types, and Use Cases. 2022. Disponível em: <https://www.altexsoft.com/blog/data-pipeline-components-and-types/>. Acesso em: 11 jun. 2024.

ATLAN. **Batch Processing vs Stream Processing: 7 Key Differences**. 2023. Disponível em: <https://atlan.com/batch-processing-vs-stream-processing/>. Acesso em: 11 jun. 2024.

AWS. **O que é um pipeline de dados?** 2022. Disponível em: <https://aws.amazon.com/pt/what-is/data-pipeline/>. Acesso em: 11 jun. 2024.

CONFLUENT. **Batch Processing vs Real Time Data Streams**. 2023. Disponível em: <https://www.confluent.io/learn/batch-vs-real-time-data-processing/>. Acesso em: 11 jun. 2024.

HAZELCAST. **What is a Data Pipeline?**2023. Disponível em: <https://hazelcast.com/glossary/data-pipeline/ >. Acesso em: 11 jun. 2024.

IBM. **O que é um pipeline de dados?** 2022. Disponível em: <https://www.ibm.com/br-pt/topics/data-pipeline>. Acesso em: 11 jun. 2024.

PALAVRAS-CHAVE

Palavras-chave: Batch. Stream. Processamento em lote.