	UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE CENTRO DE TECNOLOGIA DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO DCA0132 – ENGENHARIA DE DADOS

Projeto – Criação de pipelines de processamento de dados utilizando Apache Spark e Apache AirFlow

Entrega: 21/08/2024 23h59

Objetivo:

Desenvolver pipelines de processamento de dados em tempo real por meio da criação de aplicações Apache Spark Streaming, que serão responsáveis pelo consumo de dados a partir de um Data Lake. Utilizando a ferramenta Apache Airflow, o objetivo é agendar, orquestrar e monitorar de maneira eficiente os fluxos de processamento de dados, garantindo a escalabilidade e confiabilidade das operações ETL, com foco na análise e transformação de dados em tempo real.

Instruções:

- Trabalho a ser realizado em duplas;
- A nota deste trabalho corresponde a 100% da avaliação da Unidade III;
- A submissão do trabalho deverá ocorrer via SIGAA até a data indicada. Deverão ser submetidos os arquivos que permitam a reprodutibilidade do projeto (*docker-compose.yml* e afins), ou um link para o *github* do projeto desenvolvido;
- O projeto criado deverá ser apresentado de forma prática, com execução de exemplos, em horário a ser agendado nos dias 22/08/2024 e 23/08/2024 (conforme agenda a ser disponibilizada pelo professor, em local e formato a definir);
- No dia da apresentação, é responsabilidade das duplas preparar todos os recursos necessários para apresentar o funcionamento do projeto.

Forma de avaliação:

- Cada dupla irá apresentar e explicar o desenvolvimento do projeto e os resultados encontrados;
- O projeto será avaliado de acordo com a implantação e as soluções utilizadas para a obtenção dos resultados, sendo que no dia da apresentação o mesmo deverá estar operacional e ser apresentado o seu funcionamento. Pretende-se que no ato da apresentação não seja necessário "esperar" por instalações e compilações.

Quesitos a serem avaliados:

- Qualidade na apresentação do projeto;
- Originalidade na realização das tarefas;
- Profundidade dos detalhes abordados.

Descrição geral do projeto:

Pretende-se que seja criada uma solução de processamento de dados em tempo real utilizando Apache Spark Streaming e Apache Airflow, que deverá consumir dados em tempo-real a partir de um Data Lake. O Data Lake deve conter dados estruturados ou semi-estruturados armazenados em bancos de dados PostgreSQL ou MongoDB, bem como arquivos json e csv no sistema de arquivos local. As aplicações Spark deverão ser desenvolvidas em pySpark e deverão consumir os dados em tempo real a partir do Data Lake e realizar transformações e análises dos dados. O Apache Kafka deverá ser utilizado para ingestão e entrega de dados em tempo real para as aplicações Spark. A Figura 1 apresenta a arquitetura sugerida para o projeto. O Apache Airflow será utilizado para orquestrar a programação e o monitoramento dos fluxos de ETL do projeto, garantindo a execução confiável e escalável das tarefas de processamento de dados em tempo real.

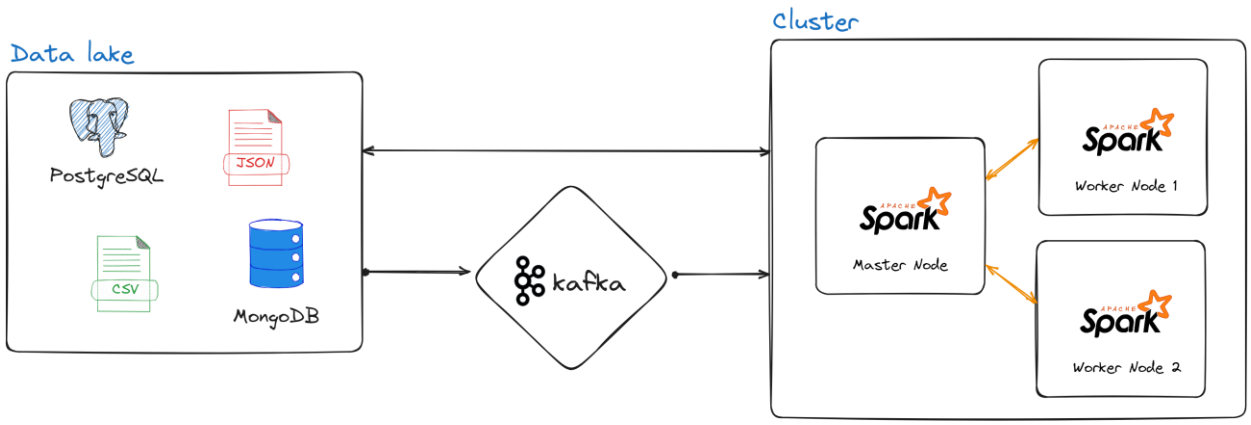

Figura 1 – Arquitetura sugerida para o projeto.

Objetivos específicos do projeto:

- Criação de um Data Lake combinando diversas fontes de dados para consumo por parte das aplicações Apache Spark a serem desenvolvidas. A escolha do conjunto de dados (dataset) é livre.
- Criação de fluxos de processamento de dados (streaming) para consumir e processar dados em tempo real com Apache Spark a partir das fontes de dados existentes no Data Lake, com agendamento e orquestração feitos pelo Apache Airflow.
- Realizar uma análise simplificada dos dados que demonstrem o funcionamento dos pipelines criados, com o suporte do Apache Airflow para automatizar a geração e a entrega dessas análises.
- (Opcional) Criação de fluxos de migração de dados dos bancos de dados PostgreSQL para MongoDB e vice-versa, por meio de aplicações Apache Spark Streaming, integradas com o Apache Airflow para agendar e monitorar esses fluxos de migração.

Tarefas:

1. Preparação da Infraestrutura
   1.1. Identificar os requisitos, como os tipos de dados a serem processados, as fontes de dados, as ferramentas necessárias e os resultados esperados.
   1.2. Definir quais ferramentas devem ser utilizadas para o provisionamento da infraestrutura com base nos requisitos definidos. Recomenda-se a utilização da ferramenta Docker, com auxílio do docker-compose.

2. Criação do Data Lake
   2.1. Definir uma estrutura para o Data Lake, que contenha bancos de dados PostgreSQL ou MongoDB, bem como arquivos armazenados em (escolher um dos seguintes): um sistema de arquivos distribuído, como o HDFS, ou no sistema de arquivos local do sistema operacional.
   2.2. Escolher um (ou alguns) conjunto(s) de dado(s) para ser(em) armazenado(s) no Data Lake. Pretende-se que dados sejam armazenados nos bancos de dados mencionados, bem como em arquivos do tipo json e csv no sistema de arquivos definido (se necessário).
   2.3. Ao escolher os conjuntos de dados para armazenar no Data Lake, priorize conjuntos de dados relevantes para o projeto e que possam ser utilizados para demonstrar o funcionamento e validar os pipelines criados.
3. Criação do Cluster
   3.1. Instalar o Apache Spark na versão 3.5+ (compilação Scala 2.12), com no mínimo 2 nós, sendo um mestre e um *worker*.
   3.2. Realizar as configurações necessárias nos nós para compor o cluster de processamento dos dados.
4. Configuração do Kafka:
   4.1. Instalar e configurar o Apache Kafka para permitir a ingestão de dados em tempo real a partir de diferentes fontes.
   4.2. Desenvolver conectores para coletar dados dos bancos de dados PostgreSQL ou MongoDB.
   4.3. Utilizar os conectores desenvolvidos para realizar a ingestão de dados no Kafka, permitindo a captura em tempo real de eventos e mudanças nas fontes de dados.
5. Processamento dos Dados:
   5.1. Criar aplicações Apache Spark Streaming para processar os dados em tempo real, consumindo dados (json e csv) a partir do sistema de arquivos definido, bem como consumindo as informações dos tópicos do Kafka, realizando transformações e análises dos resultados do processamento.
   5.2. As aplicações criadas devem ser desenvolvidas utilizando em linguagem Python (pySpark) e executadas/processadas no cluster criado.
6. Armazenamento dos Dados
   6.1. Salvar os dados processados no Data Lake, seja no banco de dados PostgreSQL ou MongoDB, a depender da aplicação.
   6.2. (Opcional) Desenvolver processos de ETL para carregar os dados do MongoDB para o PostgreSQL e do PostgreSQL para o MongoDB.
7. Orquestração dos Fluxos de ETL com Apache Airflow
   7.1. Implementar o Apache Airflow para criar, agendar e monitorar os fluxos de processamento de dados (ETL). Configure tarefas do Apache Airflow para executar as etapas do ETL definidas nas etapas anteriores, garantindo que o processamento de dados seja executado de forma escalonável, confiável e automatizada.
   7.2. Desenvolver DAGs (Directed Acyclic Graphs) no Apache Airflow, onde cada DAG representará um fluxo de ETL específico. Defina as dependências entre as tarefas para garantir que o processamento seja executado na ordem correta e que os fluxos de dados sejam gerenciados eficazmente.
   7.3. Estabeleça um agendamento apropriado para os DAGs do Apache Airflow com base nos requisitos de frequência de processamento de dados, garantindo que as operações de ETL ocorram no momento adequado.
   7.4. (Opcional) Crie painéis de controle ou painéis de monitoramento para acompanhar o status e o desempenho dos fluxos de ETL executados pelo Apache Airflow, permitindo uma visão geral clara do processo de processamento de dados.