PUC Minas

Apache Spark

Diego Roberto Gonçalves de Pontes

O que é Apache Spark?

- Apache Spark é um framework de processamento paralelo de dados, open-source, projetado para realizar processamento de dados em larga escala.
- Desenvolvido originalmente na UC Berkeley's AMPLab, Spark se tornou um dos projetos de maior sucesso da Apache Software Foundation.
- Capaz de processar grandes volumes de dados, tanto em modo batch quanto em tempo real.
- Suporta múltiplas linguagens de programação como Java, Scala, Python e R.

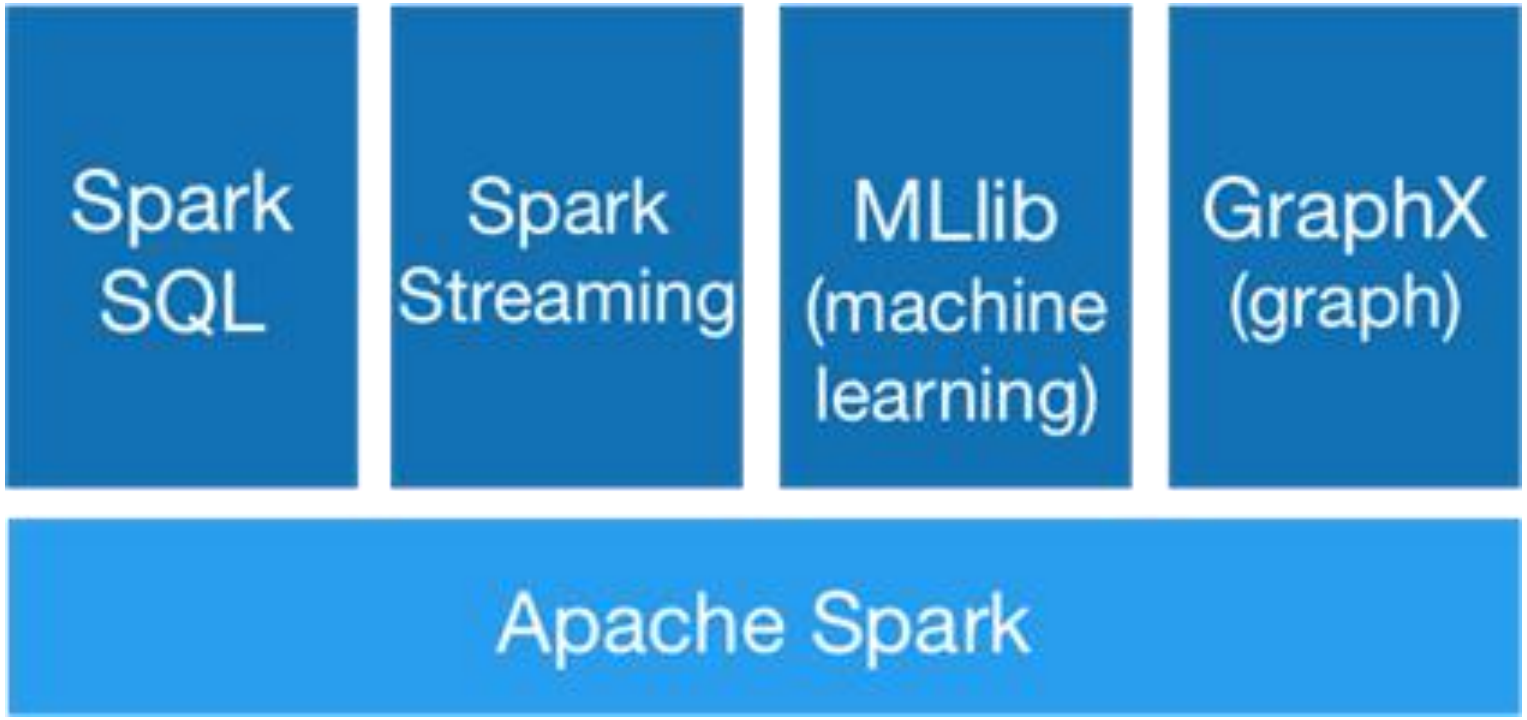
Arquitetura Spark

- Driver:
  - Responsável por criar e gerenciar o contexto Spark.
  - Distribui tarefas aos executores.
- Cluster Manager:
  - Aloca recursos entre todos os aplicativos no cluster.
    - Exemplos incluem Hadoop YARN, Apache Mesos, e Standalone Scheduler.

Arquitetura Spark

- Executors:
  - Processos que executam tarefas atribuídas pelo driver.
  - Armazenam dados em memória para processamento rápido.
- Tasks:
  - Menores unidades de trabalho que o driver distribui aos executores.
  - Cada tarefa é parte de uma operação maior.

Componentes do Spark



Resilient Distributed Datasets

- Resilient Distributed Datasets (RDD) é uma estrutura de dados fundamental do Spark. É uma coleção distribuída imutável de objetos. Cada conjunto de dados no RDD é dividido em partições lógicas, que podem ser computadas em diferentes nós do cluster. Os RDDs podem conter qualquer tipo de objeto Python, Java ou Scala, incluindo classes definidas pelo usuário.

Resilient Distributed Datasets

- RDDs são a unidade fundamental de dados em Spark. São imutáveis.
- Resilient: se dados na memória são perdidos, podem ser recriados.
- Distributed: armazenados na memória por todo o cluster.
- Datasets: dados iniciais podem vir de um arquivo ou ser criado programaticamente.

Resilient Distributed Datasets

- 2 tipos de operações, de transformação ou de ação.
- Exemplos de operações de transformação.
  - map(function) -> cria um novo RDD processando a função em cada registro do RDD.
  - filter(function) -> cria um novo RDD incluindo ou excluindo cada elemento de acordo com um função booleana.
  - outros: distinct, sample, union, intersection, subtract, cartesian, combineByKey, groupByKey, join, etc.
- Exemplos de operações de ações.
  - count() -> retorna o número de elementos.
  - take(n) -> retorna um arraycom os primeiros n elementos.
  - collect() -> retorna um arraycom todos os elementos.
  - saveAsTextFile(file) -> salva o RDD no arquivo.

Lazy Evaluation

- Lazy Evaluation significa que as operações em RDDs não são executadas imediatamente quando são definidas. Em vez disso, elas são registradas e apenas quando uma ação é chamada, o Spark executa todas as transformações necessárias de uma vez. Isso permite ao Spark otimizar o plano de execução e minimizar o movimento de dados, resultando em um processamento mais eficiente.

Exemplo Lazy Evaluation

- Imagine o gerenciamento de uma biblioteca que precisa realizar várias tarefas relacionadas aos livros. No entanto, o bibliotecario decide não realizar as tarefas imediatamente, mas sim anotar cada tarefa que precisa ser feita. O bibliotecario só executa essas tarefas quando alguém realmente precisa dos resultados.

Exemplo

- Coleção inicial:
  - O bibliotecário tem uma lista de todos os livros da biblioteca
- Tarefas anotadas:
  - Tarefa 1: Filtrar os livros que foram publicados após o ano 2000.
  - Tarefa 2: Multiplicar por dois o número de páginas de cada um desses livros filtrados.
  - Tarefa 3: Manter apenas os livros cujo número de páginas multiplicado por dois é maior que 300.
- Execução da tarefa:
  - Alguém chega na biblioteca e pede a lista de livros que atenda aos critérios; somente assim o bibliotecário vai executar as tarefas.

PUC Minas