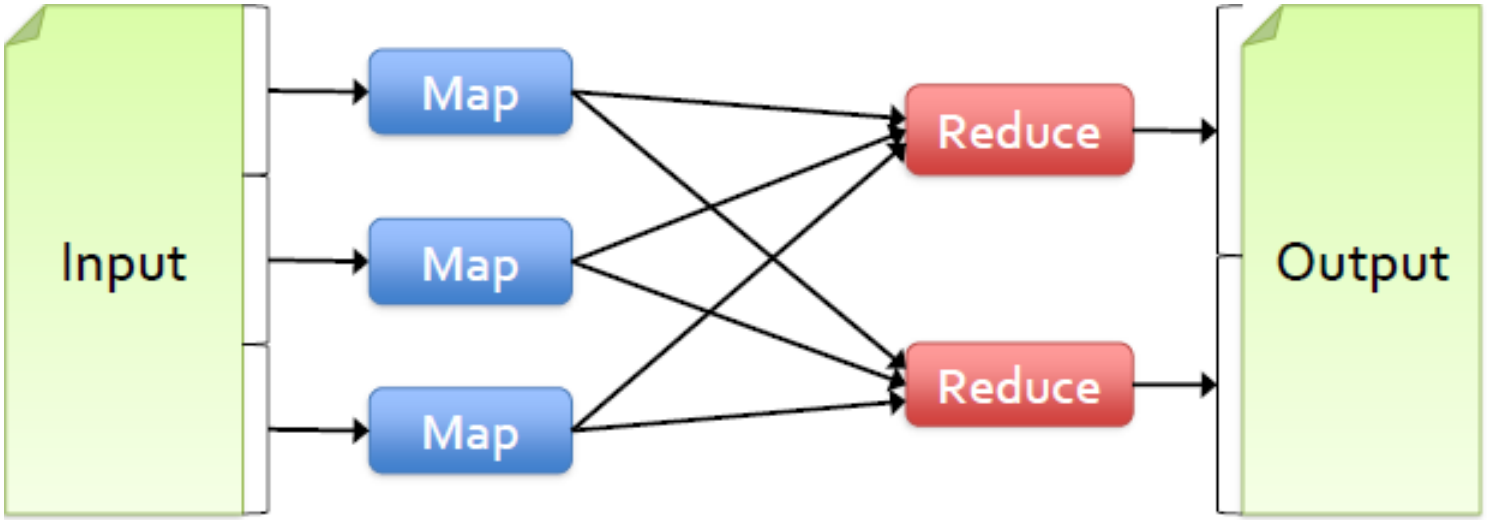
Spark

Motivação

- Os modelos de programação populares atuais para clusters transformam dados que fluem de armazenamento estável para armazenamento estável
- por exemplo, MapReduce:

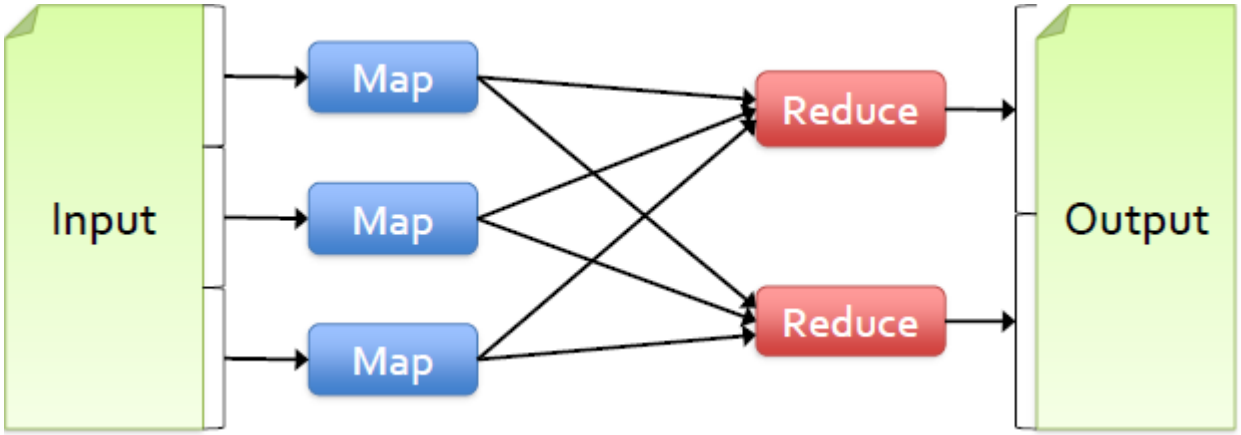


Definição

- MapReduce: é um modelo de programação para suportar computações paralelas em grandes coleções de dados em clusters de computadores. O MapReduce passa a ser considerado um novo modelo computacional distribuído, inspirado pelas funções map e reduce. MapReduce é um “Data-Oriented” que processa dados em duas fases primárias: Map e Reduce.
- Exemplo Hadoop

Definição

- Função Map: ou função de pareamento, processa os dados de entrada, criando um par chave-valor. Se a função estiver fazendo contagem de palavras de um texto, a saída será o par chave-valor, palavra e o número de ocorrências.
- Função Reduce: a função toma todas as saídas da função map e reduz a saídas únicas.



Background

- Clusters de commodities tornaram se uma importante plataforma de computação para uma variedade de aplicações Na indústria: search , tradução automática, segmentação de anúncios, etc
- Em pesquisa: bioinformática, NLP, simulação climática, etc
- Modelos de programação de cluster de alto nível, como o MapReduce , suportam muitas dessas aplicações

Limitações da computação em larga escala

- Historicamente , a computação estava ligada ao processador

–O volume de dados era relativamente pequeno

–Cálculos complicados são realizados sobre os dados

- Os avanços na tecnologia de computação centraram se historicamente em torno da melhoria do poder de uma única máquina

Avanços em CPUs

- Moore's Law

–O número de transistores em um circuito integrado duplicava a cada dois anos

- A computação em um single core não pode escalar com as necessidades de computação atuais

Limitação do single core

- O consumo de energia limita o aumento de velocidade que se obtém do transistor

Sistemas distribuídos

Permite aos desenvolvedores

usar múltiplas máquinas

para uma mesma tarefa

Problemas com Sistemas Distribuídos

- A programação em um sistema distribuído é muito mais complexa

–Sincronizar trocas de dados

–Gerenciar uma largura de banda finita

- Controlar o tempo de computação é complicado

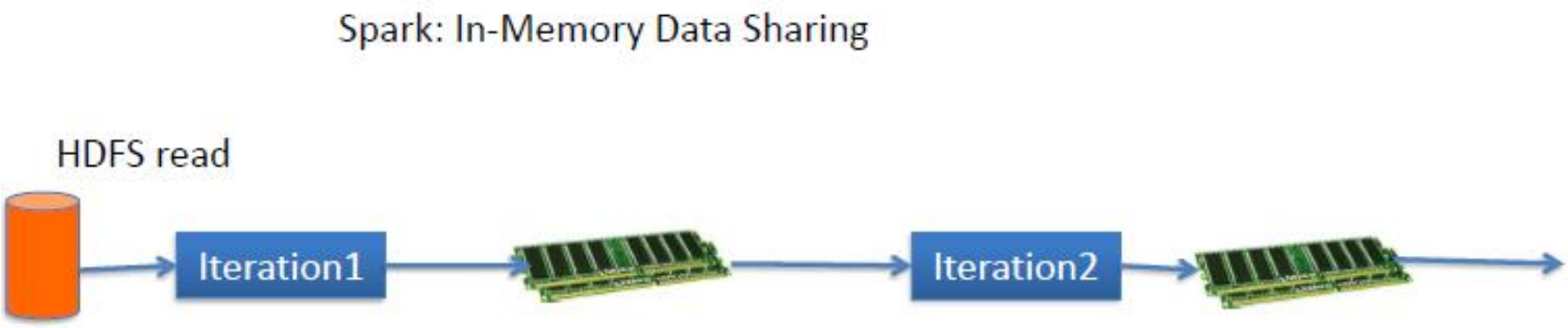
“You know you have a distributed system when the crash of a computer you’ve never heard of stops you from getting any work done.”
Leslie Lamport

-Os sistemas distribuídos devem ser projetados com a expectativa de falha

Apache Spark

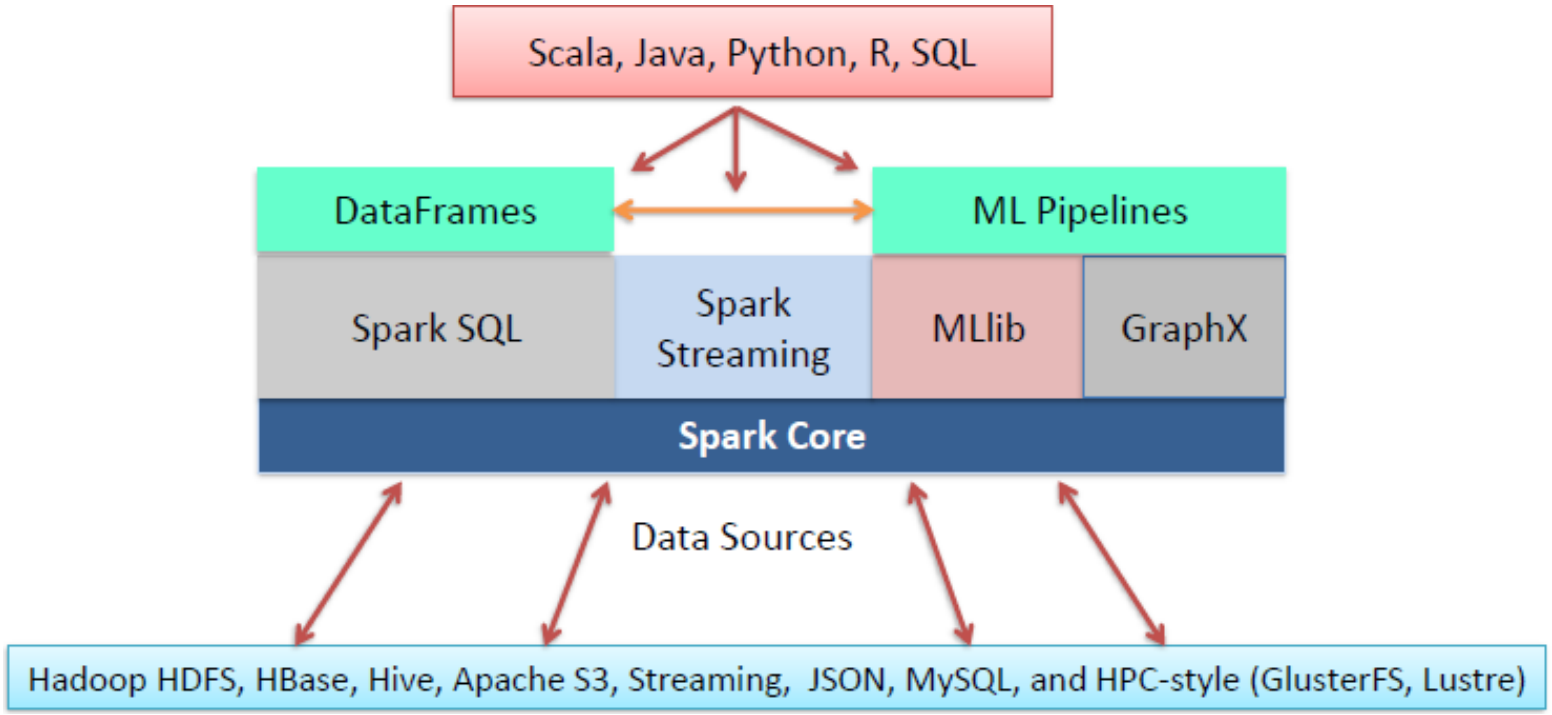
- Engine de processamento; em vez de apenas “map” e “reduce “, define um grande conjunto de operações (transformações e ações)
- As operações podem ser arbitrariamente combinadas em qualquer ordem
- Software livre
- Suporta Java, Scala, Python e R

Apache Spark



Apache Spark

- Apache Spark suporta data analysis, machine learning, graphs, streaming data, etc.
- É capaz de ler/escrever vários tipos de dados e permite desenvolvimento em várias linguagens



Resilient Distributed Datasets (RDDs)

- Elemento chave: Resilient Distributed Dataset (RDD)

- Consiste na partição e distribuição dos dados para diversas máquinas ou cores mantendo a consistência dos dados.

Resilient Distributed Datasets (RDDs)

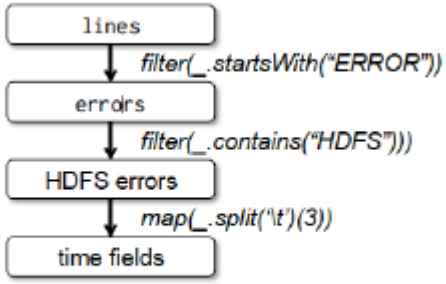
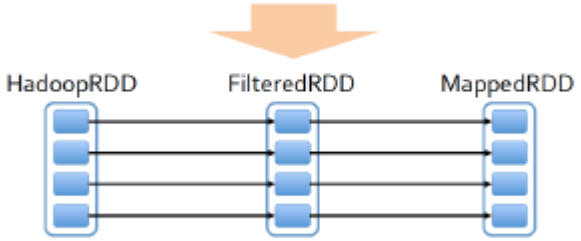


Figure 1: Lineage graph for the third query in our example. Boxes represent RDDs and arrows represent transformations.

RDDs track the graph of transformations that built them (their *lineage*) to rebuild lost data

E.g.: `messages = textFile(...).filter(_.contains("error")).map(_.split('\t')(2))`



https://www.usenix.org/sites/default/files/conference/protected-files/nsdi_zaharia.pdf

https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf

Resilient Distributed Datasets (RDDs)

- As transformações geram um novo dataset a partir de um existente. Não executa a operação, apenas as configuram (avaliação preguiçosa – Lazy operation)

- As ações podem ser aplicadas a RDDs ; ações forçam cálculos e valores de retorno

Operações

Transformations (create a new RDD)	map filter sample groupByKey reduceByKey sortByKey intersection	flatMap union join cogroup cross mapValues reduceByKey
Actions (return results to driver program)	collect Reduce Count takeSample take lookupKey	first take takeOrdered countByKey save foreach

Resumo

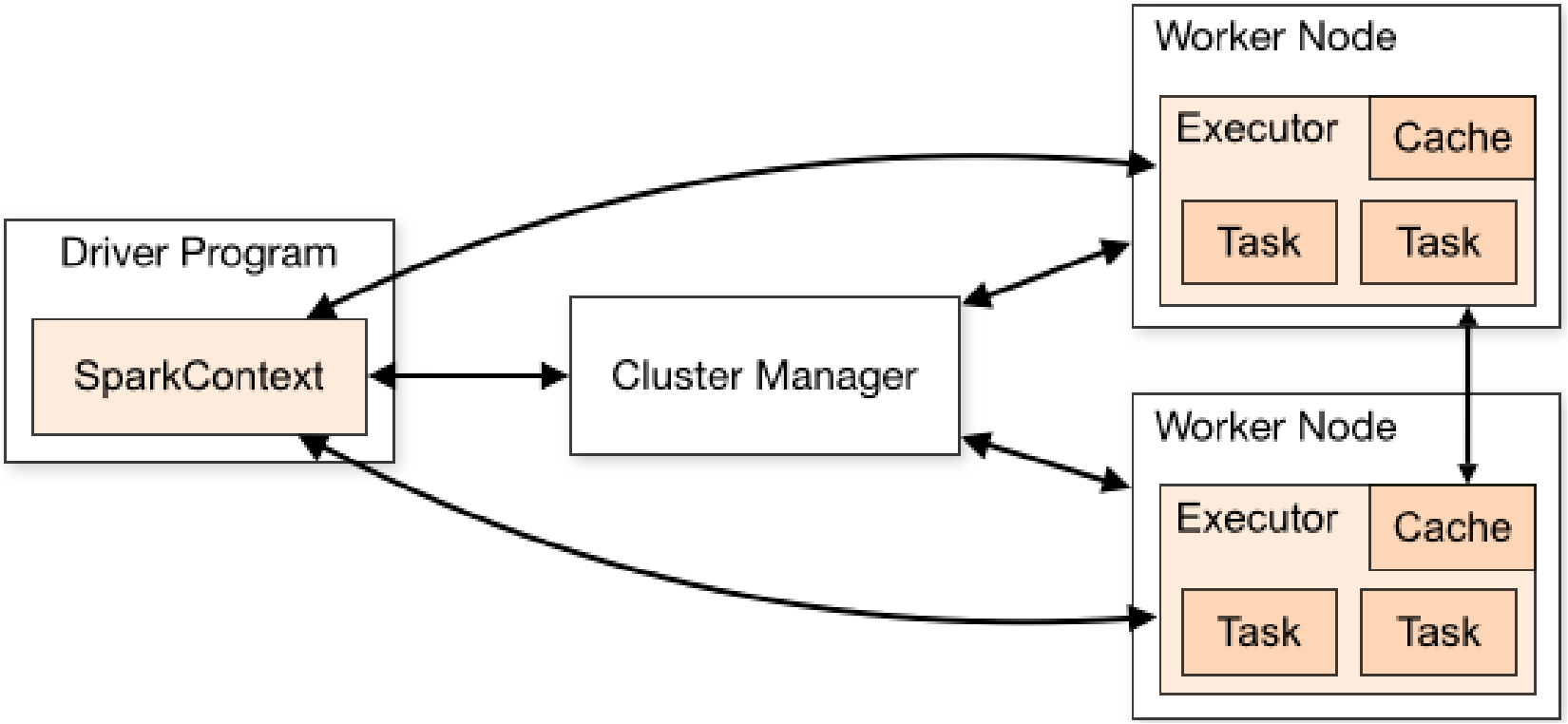
- Spark:

Plataforma para processamento de dados em larga escala.
Baseado em transformações “preguiçosas” dos dados.
Estende a abstração do Map Reduce

- Propriedades RDD:

Imutáveis (apenas leitura)
Distribuídos / Particionados
Criados de Dados Estáveis ou outra RDD

Arquitetura Spark



Arquitetura Spark

- Cluster manager é responsável por gerenciar os workers. Envia dados, tarefas a executar sobre os dados, recebe os resultados da computação distribuída, compilando em uma única saída
- O Spark só não suporta a falha no cluster manager, que atua como um worker node quando necessário.

Arquitetura Spark

- O desempenho do Spark depende fortemente das ações que são feitas sobre os dados. Caso uma operação precise ser feita observado a totalidade dos dados, não somente a partição que está sob o processamento de um único worker, ocorre degradação severa.