Programa de Aceleração Banco Inter

Processamento Spark

data sprints
we enjoy sprinting

28jan2021
www.datasprints.com.br

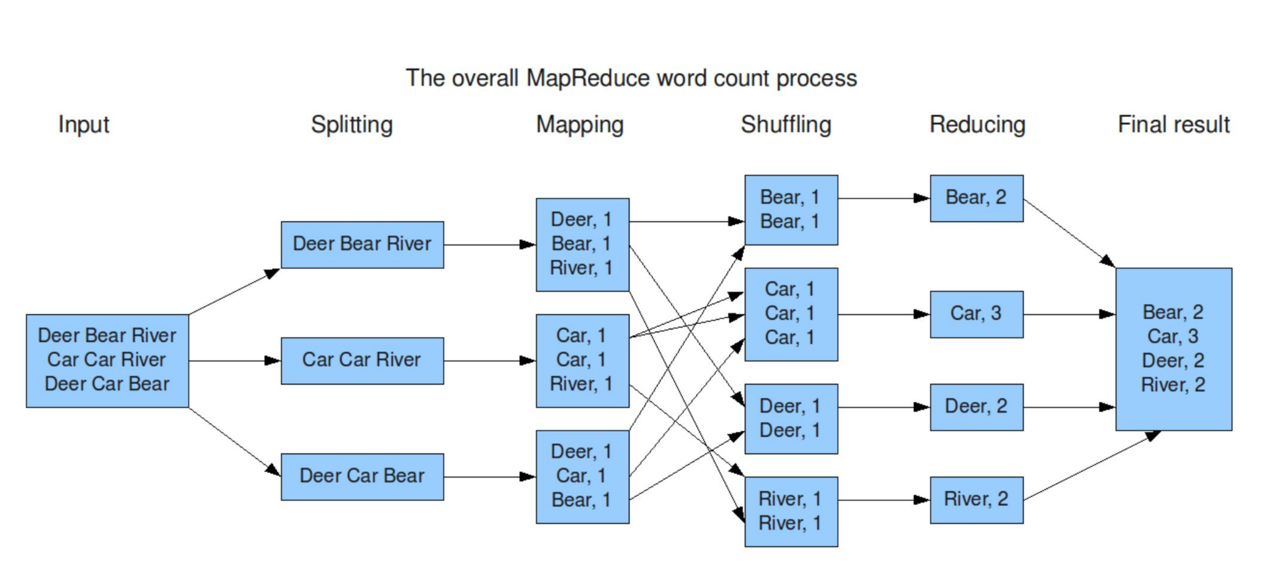
AGENDA

1. Map Reduce;
2. Intro;
3. Architecture
4. Spark features
5. Processing framework
6. Components of Spark

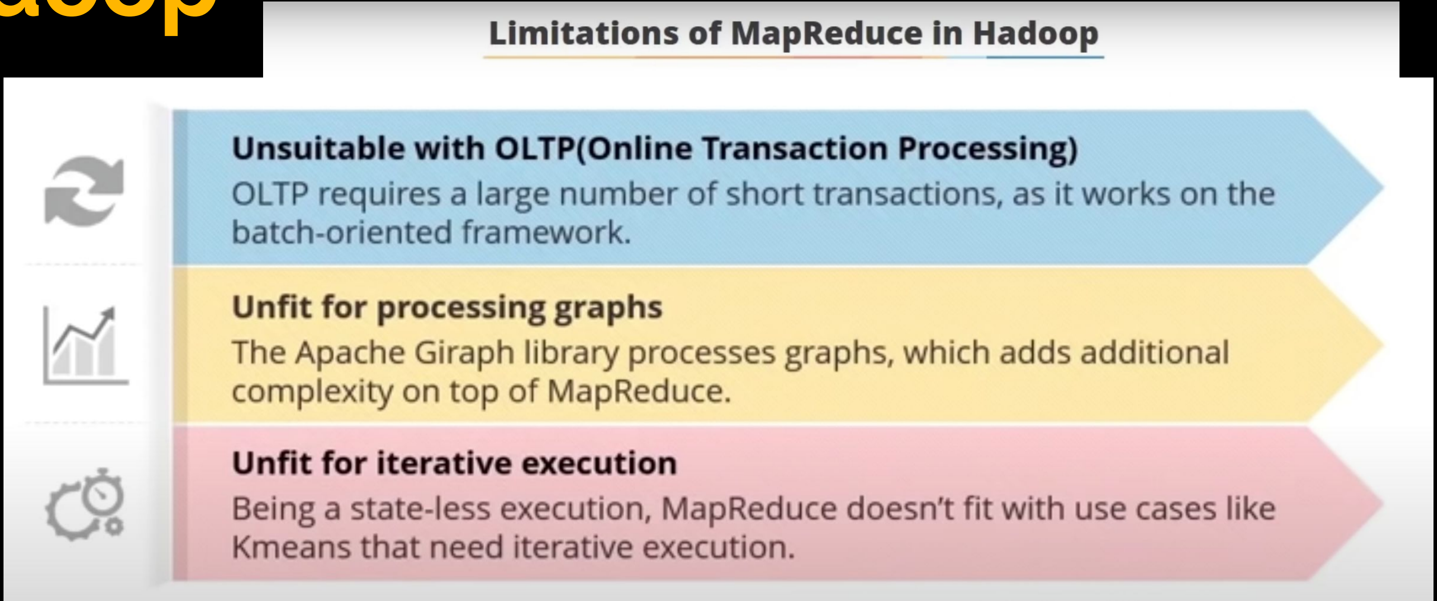
Map Reduce

MapReduce is a programming model and an associated implementation for processing and generating large data sets.
Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

Map Reduce


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2

Limitation Map Reduce on Hadoop



Limitations of MapReduce in Hadoop

Unsuitable with OLTP(Online Transaction Processing)
OLTP requires a large number of short transactions, as it works on the batch-oriented framework.

Unfit for processing graphs
The Apache Giraph library processes graphs, which adds additional complexity on top of MapReduce.

Unfit for iterative execution
Being a state-less execution, MapReduce doesn't fit with use cases like Kmeans that need iterative execution.

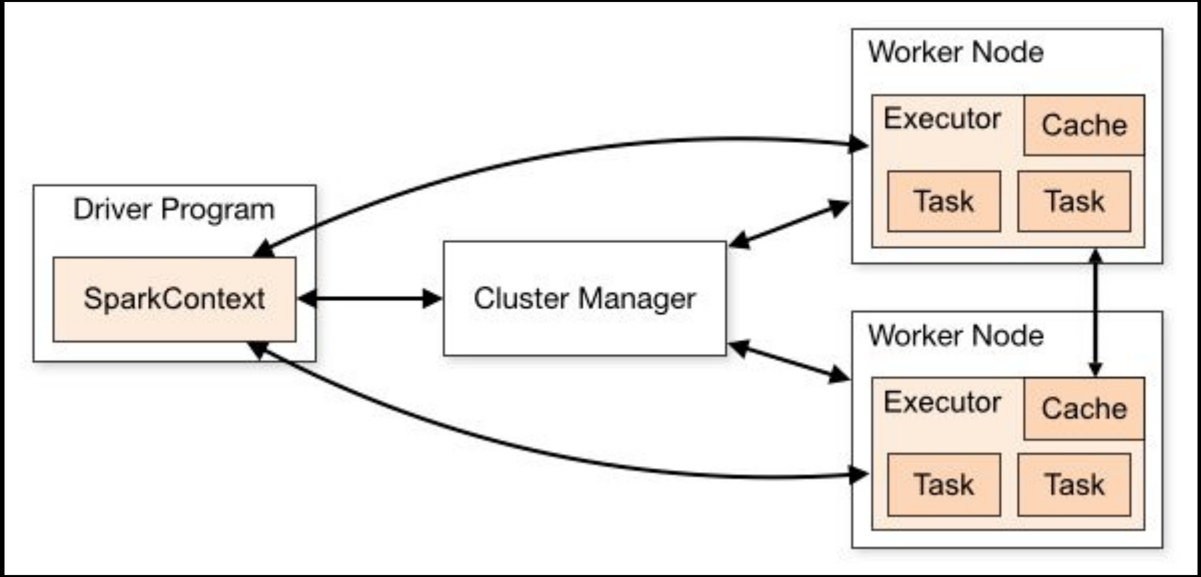
Introduction

Apache Spark is an Open source analytical processing engine for large scale powerful distributed data processing and machine learning applications. Spark is Originally developed at the University of California, Berkeley's, and later donated to Apache Software Foundation. In February 2014, Spark became a Top-Level Apache Project and has been contributed by thousands of engineers and made Spark as one of the most active open-source projects in Apache.

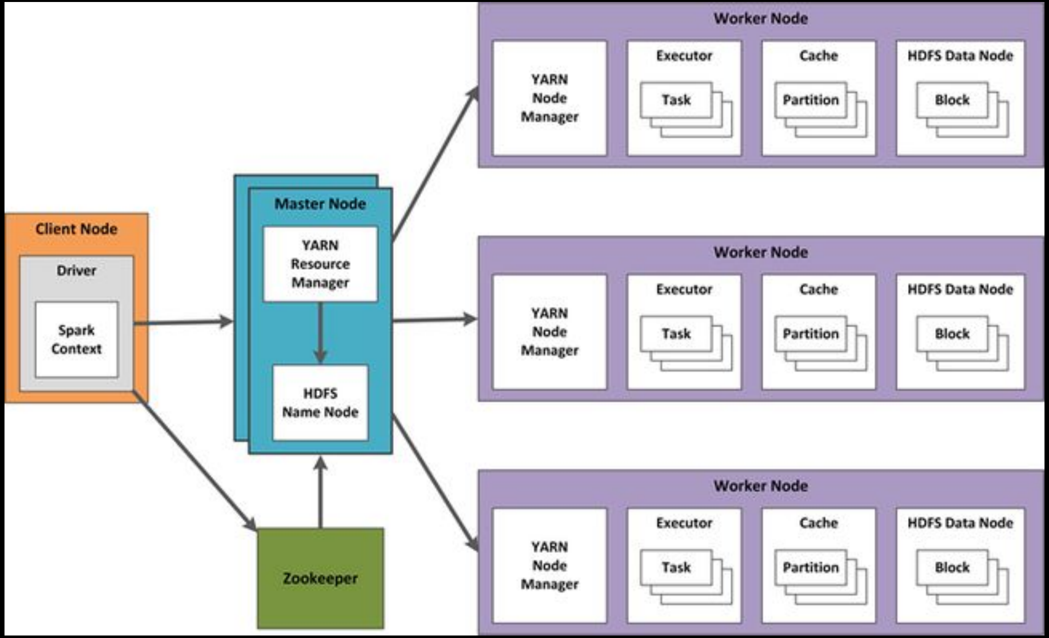
Introduction

Apache Spark is a lightning-fast cluster computing designed for fast computation. It was built on top of Hadoop MapReduce and it extends the MapReduce model to efficiently use more types of computations which includes Interactive Queries and Stream Processing.

Architecture



Architecture



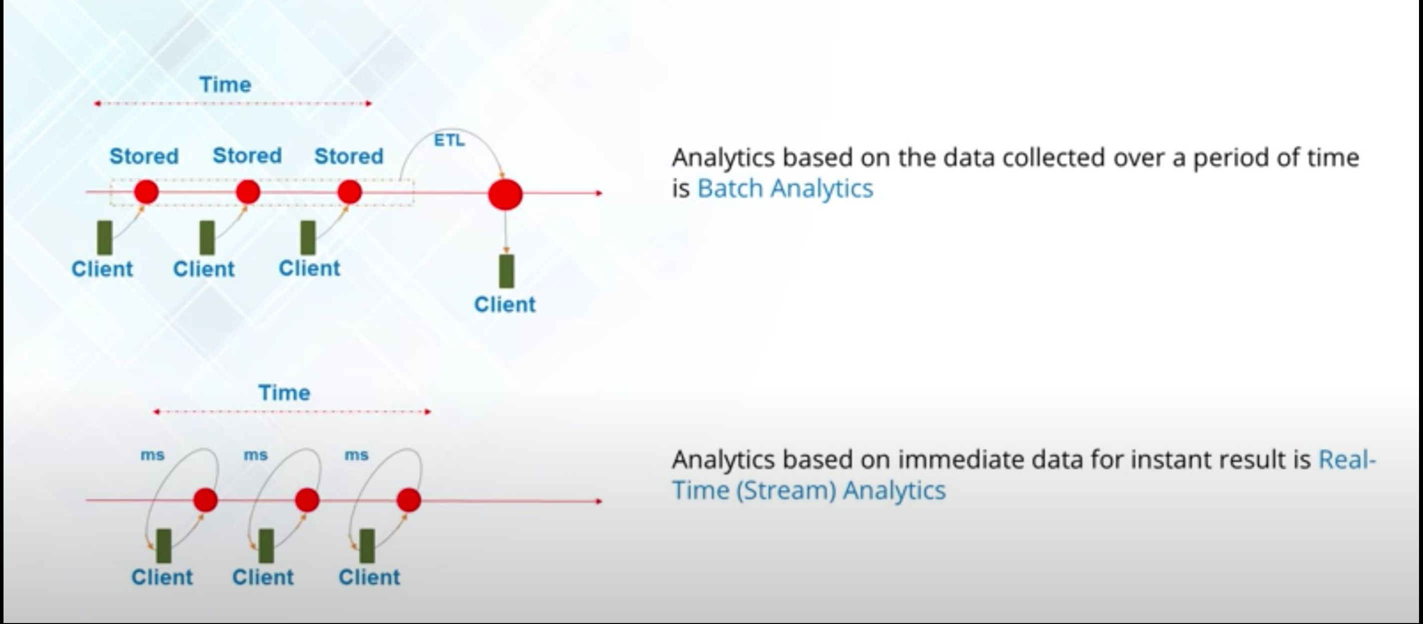
Features Spark

Speed – Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory.

Supports multiple languages – Spark provides built-in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages. Spark comes up with 80 high-level operators for interactive querying.

Advanced Analytics – Spark not only supports 'Map' and 'reduce'. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

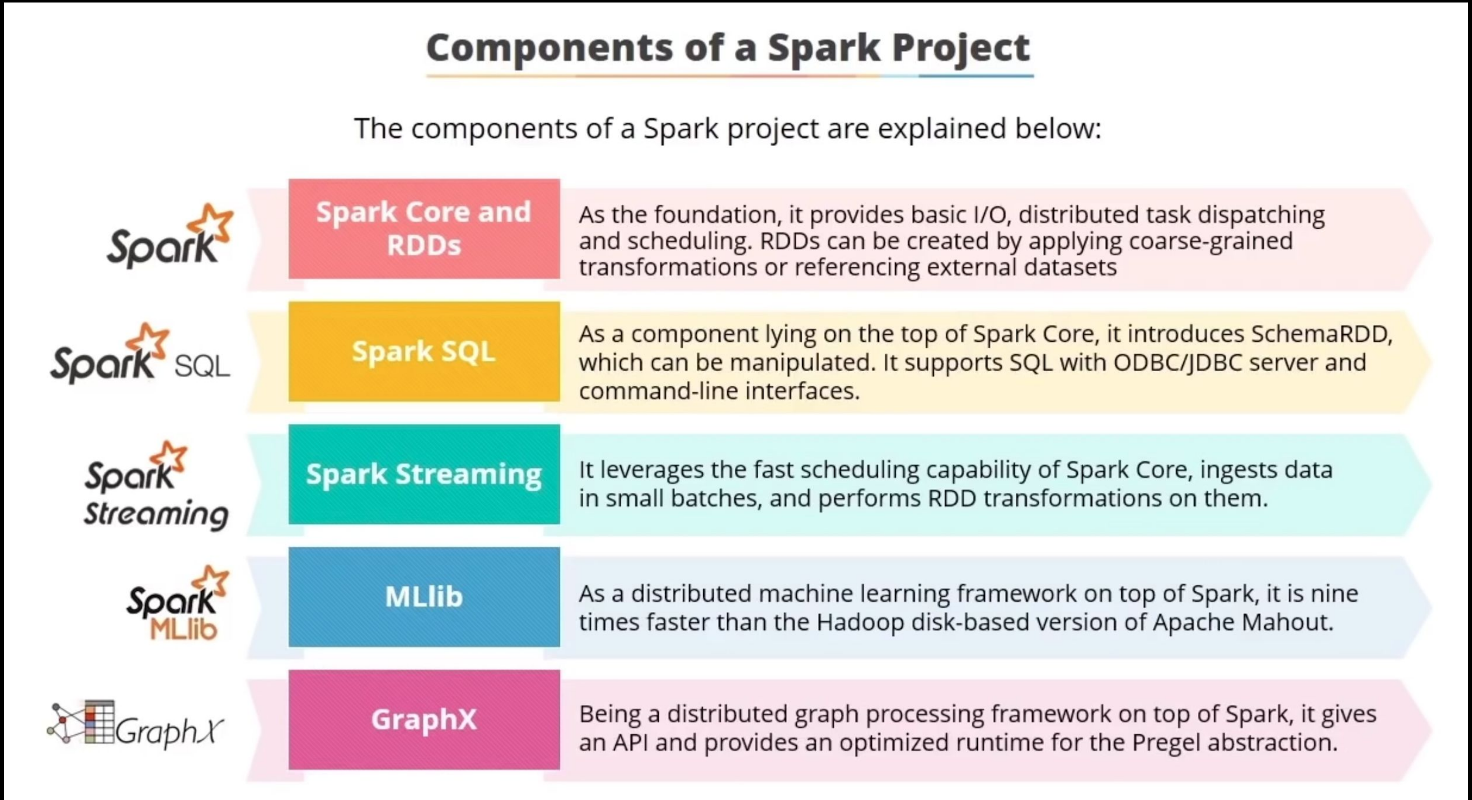
Processing Framework



Analytics based on the data collected over a period of time is Batch Analytics

Analytics based on immediate data for instant result is Real-Time (Stream) Analytics

“


Spark
Spark SQL
Spark Streaming
Spark MLlib
GraphX

Components of a Spark Project

The components of a Spark project are explained below:

Spark Core and RDDs

As the foundation, it provides basic I/O, distributed task dispatching and scheduling. RDDs can be created by applying coarse-grained transformations or referencing external datasets

Spark SQL

As a component lying on the top of Spark Core, it introduces SchemaRDD, which can be manipulated. It supports SQL with ODBC/JDBC server and command-line interfaces.

Spark Streaming

It leverages the fast scheduling capability of Spark Core, ingests data in small batches, and performs RDD transformations on them.

MLlib

As a distributed machine learning framework on top of Spark, it is nine times faster than the Hadoop disk-based version of Apache Mahout.

GraphX

Being a distributed graph processing framework on top of Spark, it gives an API and provides an optimized runtime for the Pregel abstraction.

RDD (Resilient Distributed Datasets)

Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

RDD is a read-only, partitioned collection of records. RDDs can be created through deterministic operations on either data on stable storage or other RDDs. RDD is a fault-tolerant collection of elements that can be operated on in parallel.

References

https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf
https://spark.apache.org/docs/latest/
https://www.youtube.com/watch?v=9mELEARcxJo
https://www.edureka.co/apache-spark-scala-certification-training
https://www.youtube.com/watch?v=QaoJNXW6SQo
https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm
https://sparkbyexamples.com/
https://databricks.com/spark/getting-started-with-apache-spark
https://www.udacity.com/course/learn-spark-at-udacity--ud2002

Obrigado!

data sprints

we enjoy sprinting

www.datasprints.com.br