

What is Apache Spark?

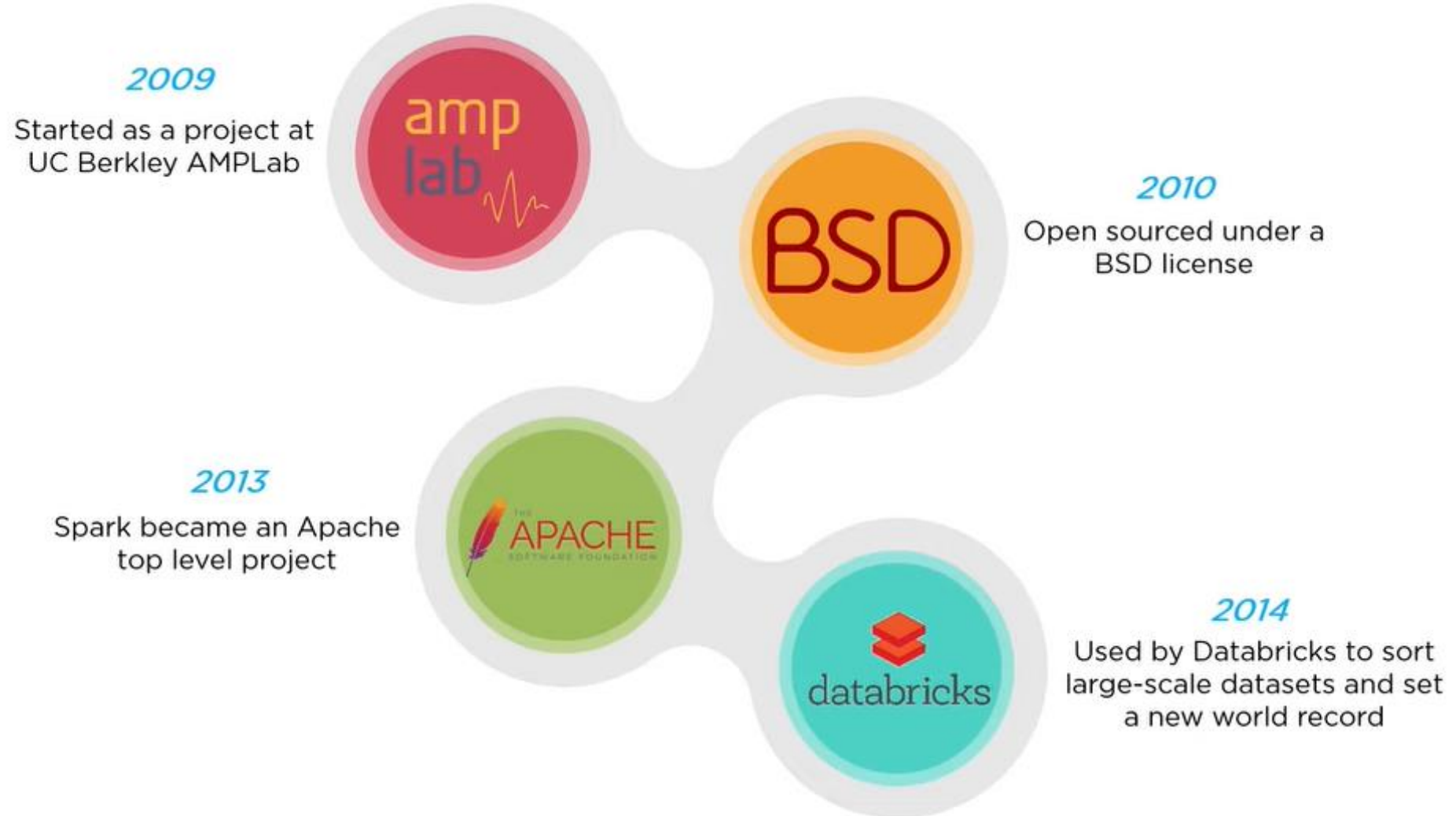


What's in it for you?

1. History of Spark
2. What is Spark?
3. Hadoop vs Spark
4. Components of Apache Spark
5. Spark Architecture
6. Applications of Spark
7. Spark Use Case



History of Apache Spark



What is Apache Spark?



Apache Spark is an open-source data processing engine to store and process data in real-time across various clusters of computers using simple programming constructs

Support various programming languages



Developers and data scientists incorporate Spark into their applications to rapidly query, analyze, and transform data at scale



Query



Analyze



Transform

Hadoop vs Spark



Hadoop vs Spark



Processing data using MapReduce in Hadoop is slow

Performs batch processing of data

Hadoop has more lines of code. Since it is written in Java, it takes more time to execute

Hadoop supports Kerberos authentication, which is difficult to manage



Spark processes data 100 times faster than MapReduce as it is done in-memory

Performs both batch processing and real-time processing of data

Spark has fewer lines of code as it is implemented in Scala

Spark supports authentication via a shared secret. It can also run on YARN leveraging the capability of Kerberos

Spark Features



Spark Features



Fast processing



Spark contains **Resilient Distributed Datasets (RDD)** which saves time taken in reading, and writing operations and hence, it runs almost ten to hundred times faster than Hadoop

Spark Features



Fast processing



In-memory computing

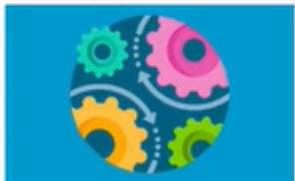


In Spark, data is stored in the **RAM**, so it can access the data quickly and accelerate the speed of analytics

Spark Features



Fast processing



**In-memory
computing**



Flexible



Spark supports **multiple languages** and allows the developers to write applications in Java, Scala, R, or Python

Spark Features



Fast processing



In-memory computing



Flexible



Fault tolerance



Spark contains [Resilient Distributed Datasets \(RDD\)](#) that are designed to handle the failure of any worker node in the cluster. Thus, it ensures that the loss of data reduces to zero

Spark Features



Fast processing



**In-memory
computing**



Flexible



Fault tolerance



Better analytics



Spark has a rich set of [SQL queries](#), [machine learning algorithms](#), [complex analytics](#), etc. With all these functionalities, analytics can be performed better

Components of Spark



Components of Apache Spark



Spark Core

Spark Core is the base engine for large-scale parallel and distributed data processing

It is responsible for:



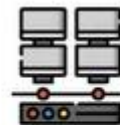
memory management



fault recovery



scheduling, distributing and
monitoring jobs on a cluster



interacting with
storage systems

APACHE
Spark

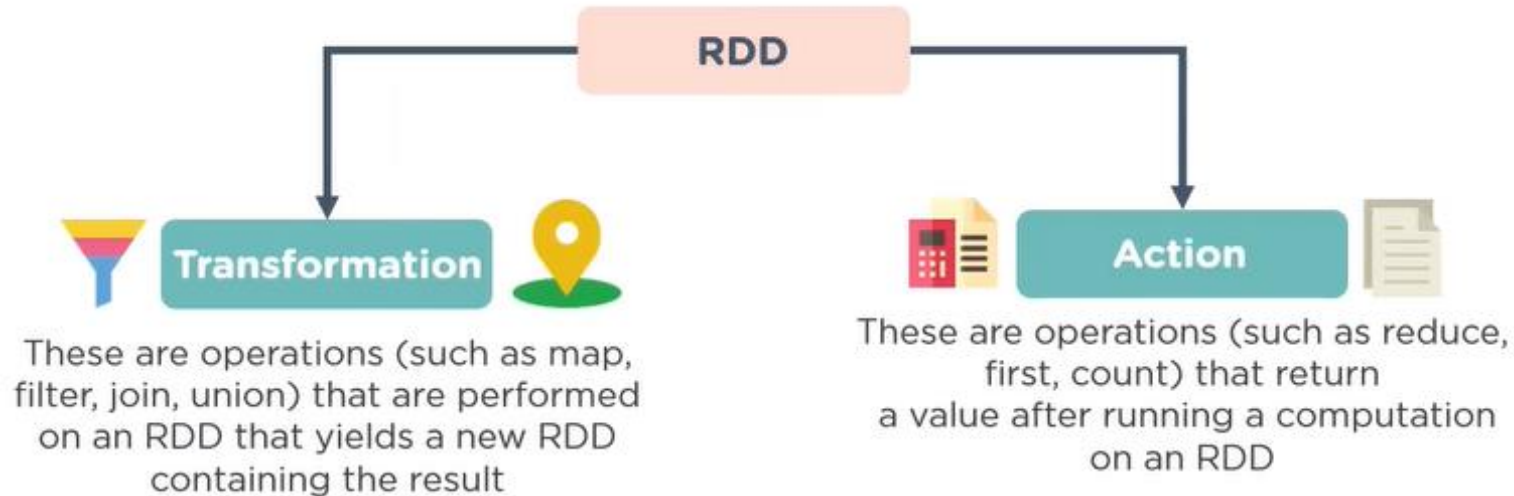
Spark Core

Resilient Distributed Dataset

Spark Core is embedded with RDDs (Resilient Distributed Datasets), an immutable fault-tolerant, distributed collection of objects that can be operated on in parallel



Spark Core



Components of Spark – Spark SQL



Spark SQL

Spark SQL framework component is used for structured and semi-structured data processing

Spark SQL Architecture

DataFrame DSL

Spark SQL and HQL

DataFrame API

Data Source API

CSV

JSON

JDBC

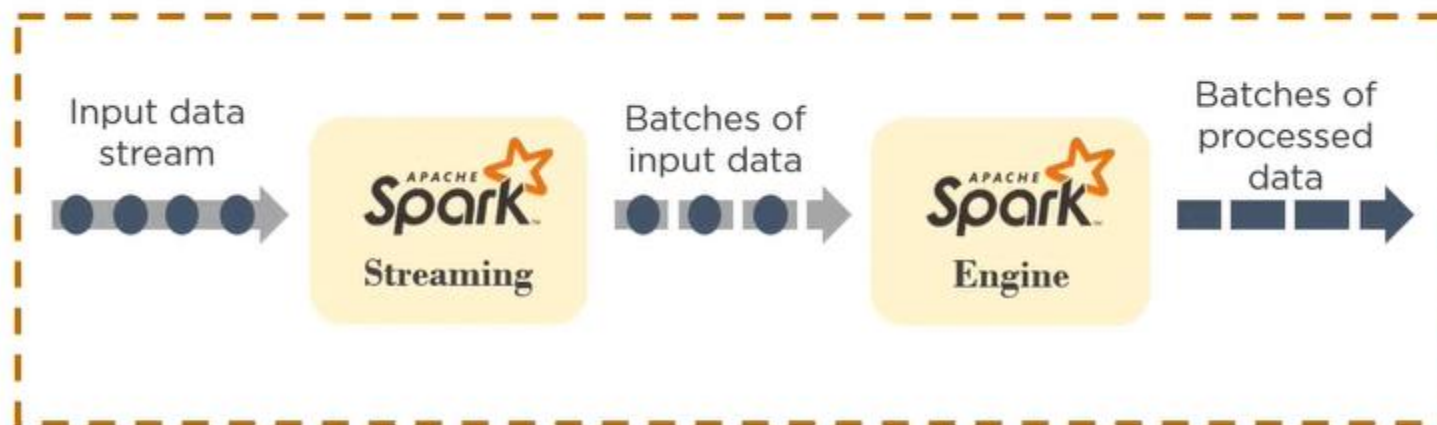


Spark SQL

Spark Streaming

Spark Streaming is a lightweight API that allows developers to perform batch processing and real-time streaming of data with ease

Provides secure, reliable, and fast processing of live data streams



Spark MLlib

MLlib is a low-level machine learning library that is simple to use, is scalable, and compatible with various programming languages

MLlib eases the deployment and development of scalable machine learning algorithms



It contains machine learning libraries that have an implementation of various machine learning algorithms



Clustering



Classification



Collaborative
Filtering

APACHE
Spark
MLlib

MLlib

GraphX

GraphX is Spark's own Graph Computation Engine and data store



Provides a uniform tool for ETL



Exploratory data analysis



Interactive graph computations

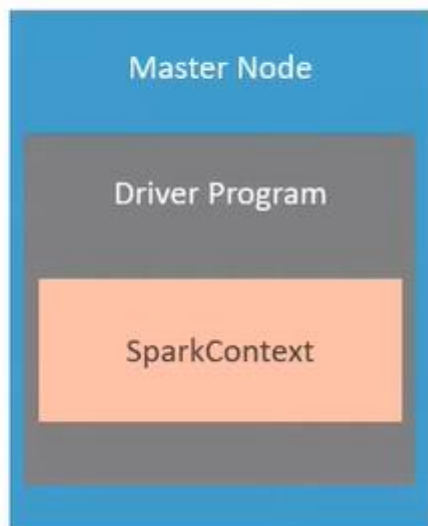


Spark Architecture



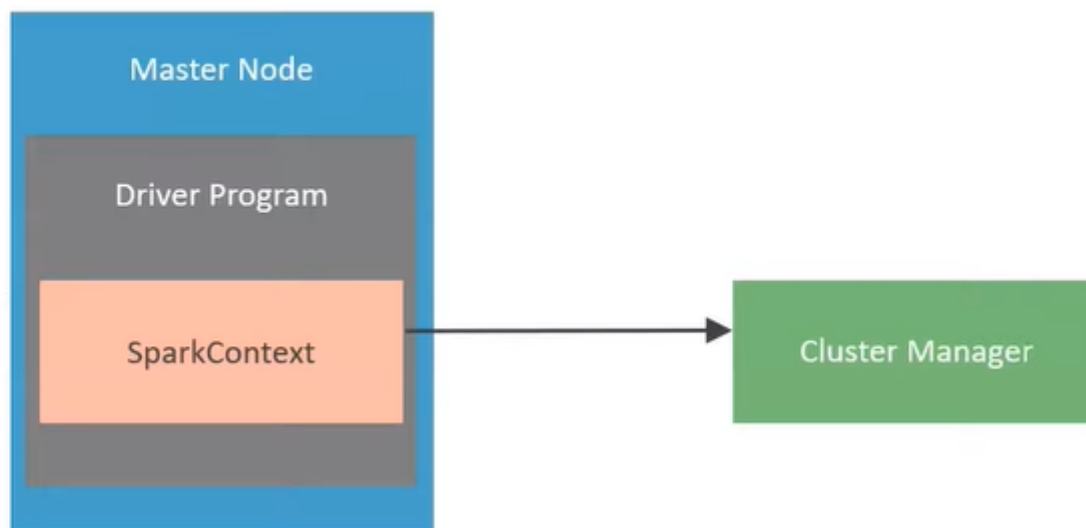
Spark Architecture

Apache Spark uses a master-slave architecture that consists of a driver, that runs on a master node, and multiple executors which run across the worker nodes in the cluster



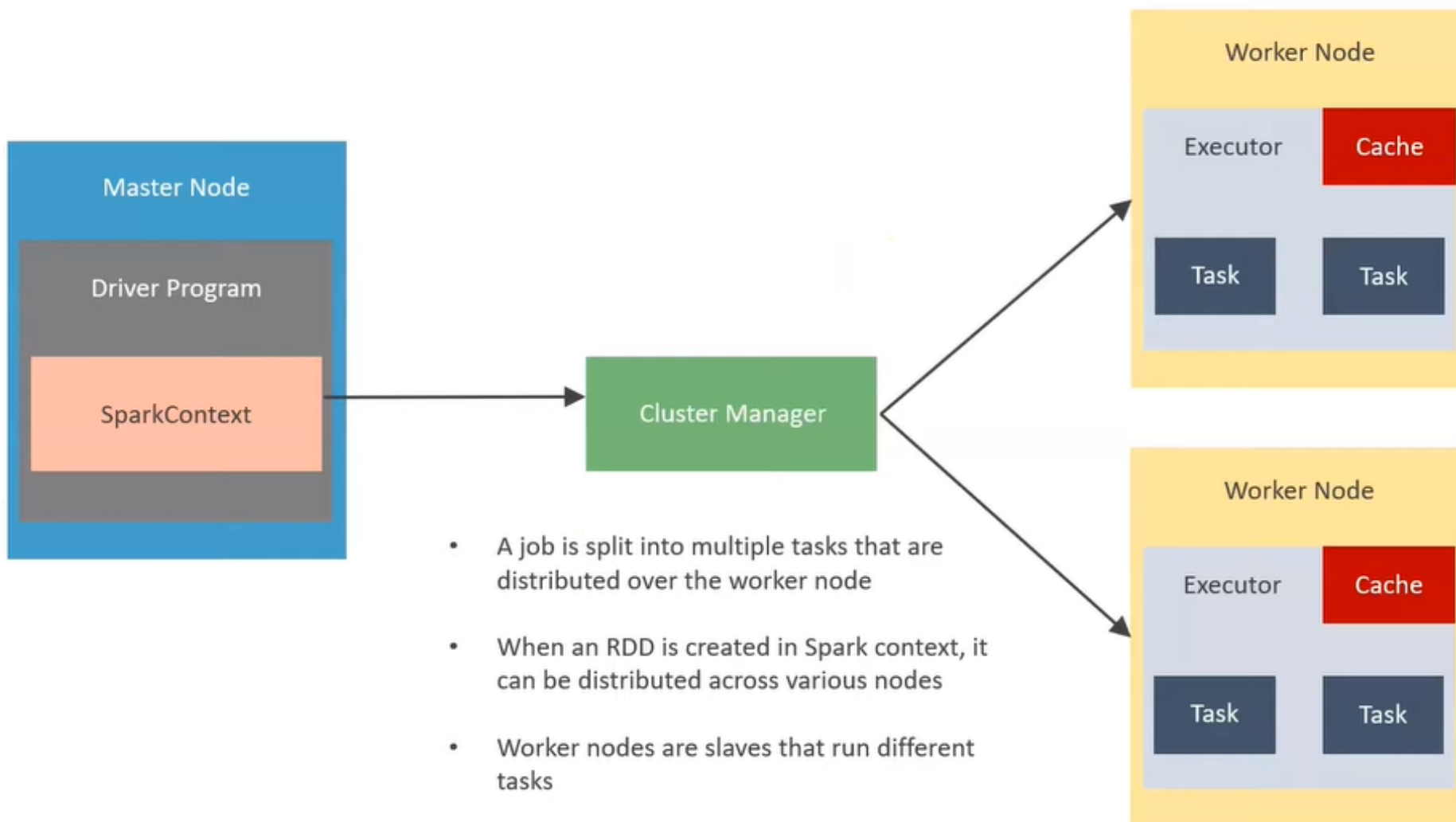
- Master Node has a Driver Program
- The Spark code behaves as a driver program and creates a SparkContext, which is a gateway to all the Spark functionalities

Spark Architecture



- Spark applications run as independent sets of processes on a cluster
- The driver program & Spark context takes care of the job execution within the cluster

Spark Architecture



Spark Cluster Managers



Standalone mode

1

By default, applications submitted to the standalone mode cluster will run in FIFO order, and each application will try to use all available nodes



MESOS

2

Apache Mesos is an open-source project to manage computer clusters, and can also run Hadoop applications



3

Apache YARN is the cluster resource manager of Hadoop 2. Spark can be run on YARN



kubernetes

4

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications

Applications of Spark



Applications of Spark



**JPMORGAN
CHASE & CO.**

JPMorgan uses Spark to detect fraudulent transactions, analyze the business spends of an individual to suggest offers, and identify patterns to decide how much to invest and where to invest

Banking



Alibaba Group

Alibaba uses Spark to analyze large sets of data such as real-time transaction details, browsing history, etc. in the form of Spark jobs and provides recommendations to its users

E-Commerce



IQVIA™

IQVIA is a leading healthcare company that uses Spark to analyze patient's data, identify possible health issues, and diagnose it based on their medical history

Healthcare



NETFLIX

**RIOT
GAMES**

Entertainment and gaming companies like Netflix and Riot games use Apache Spark to showcase relevant advertisements to their users based on the videos that they watch, share, and like

Entertainment

Spark Use Case

conviva®

Conviva is one of the world's leading video streaming companies



Video streaming is a challenge, especially with increasing demand for high-quality streaming experiences



Conviva collects data about video streaming quality to give their customers visibility into the end-user experience they are delivering

Spark Use Case

conviva®

Conviva is one of the world's leading video streaming companies



Using [Apache Spark](#), Conviva delivers a better quality of service to its customers by removing the [screen buffering](#) and learning in detail about the [network conditions](#) in real-time

This information is stored in the video player to manage live video traffic coming from [4 billion](#) video feeds every month, to ensure maximum retention

Spark Use Case

conviva®

Conviva is one of the world's leading video streaming companies



Using [Apache Spark](#), Conviva has created an auto diagnostics alert



It automatically detects [anomalies](#) along the video streaming pipeline and [diagnoses](#) the root cause of the issue



Reduces waiting time before the video starts



Avoids buffering and recovers the video from a technical error



Goal is to maximize the viewer engagement