Dell - Data Engineering Training

Uday Kumar – Data Platform Architect

Day 6

Data Engineering Training

Agenda

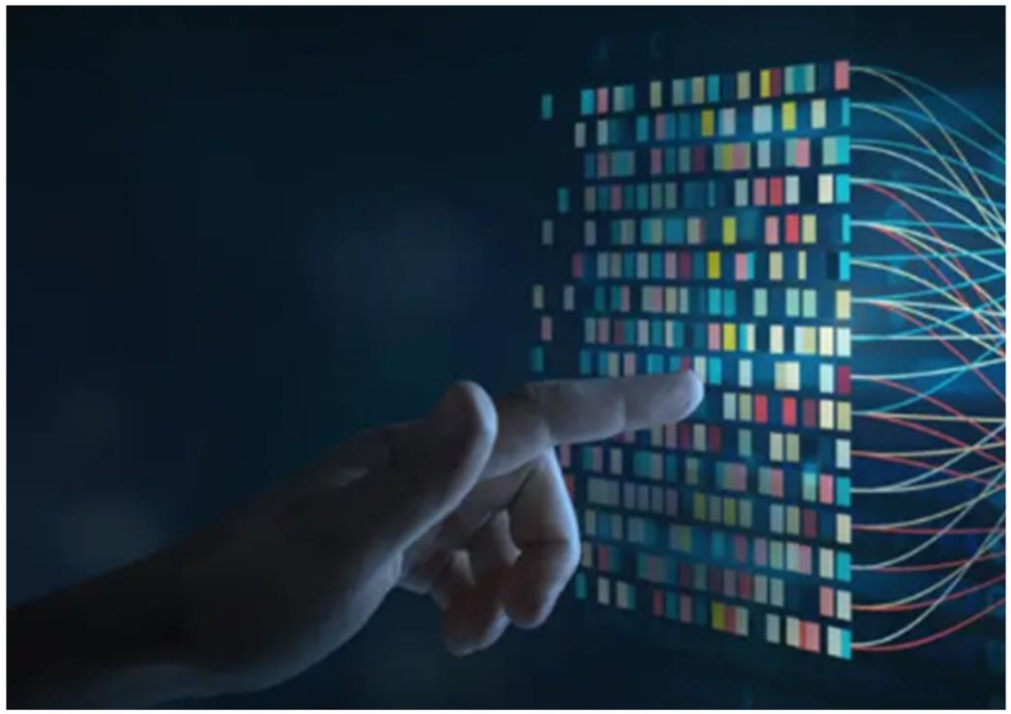
1. Data Processing with Apache Spark
2. Cluster Management and Optimization for Spark and Flink
3. Understanding Kafka
4. Files to write to Kafka Producer
5. Kafka Consumer & Stream

DATA PROCESSING WITH APACHE SPARK

Data processing is the series of operations performed on data to transform, analyze, and organize it into a useful format for further use.

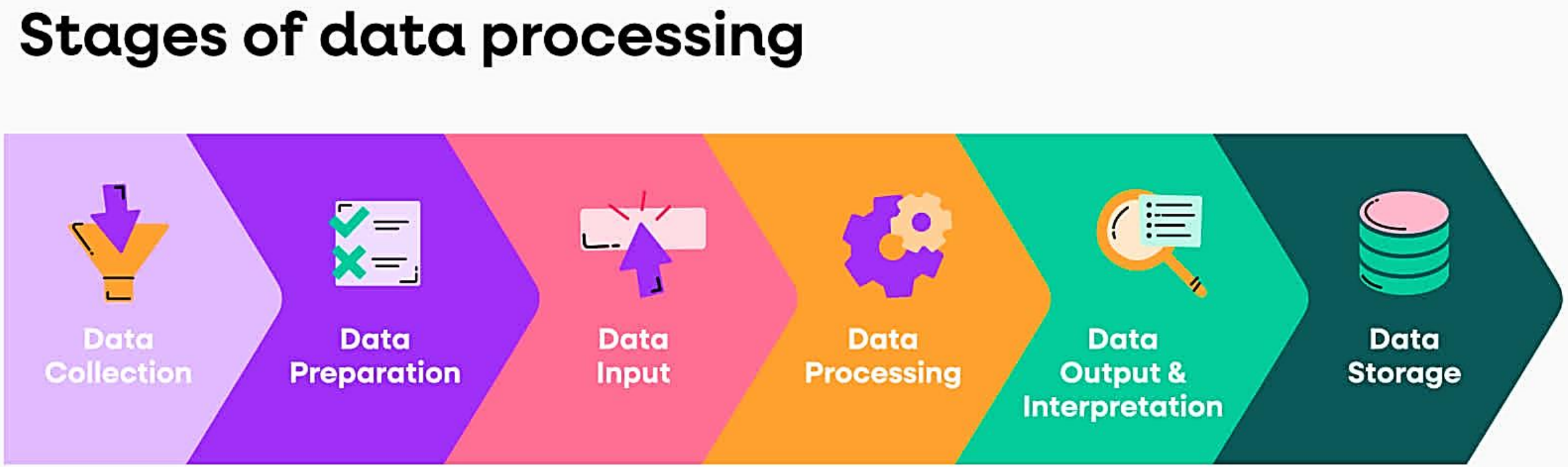
Various stages and methods are used to manipulate raw data into relevant or consumable formats. These stages often include collecting, filtering, sorting, and analyzing the data.

Data in its raw form is not useful to any organization. Data processing is the method of collecting raw data and translating it into usable information. It is usually performed in a step-by-step process by a team of data scientists and data engineers in an organization. The raw data is collected, filtered, sorted, processed, analyzed, stored, and then presented in a readable format.



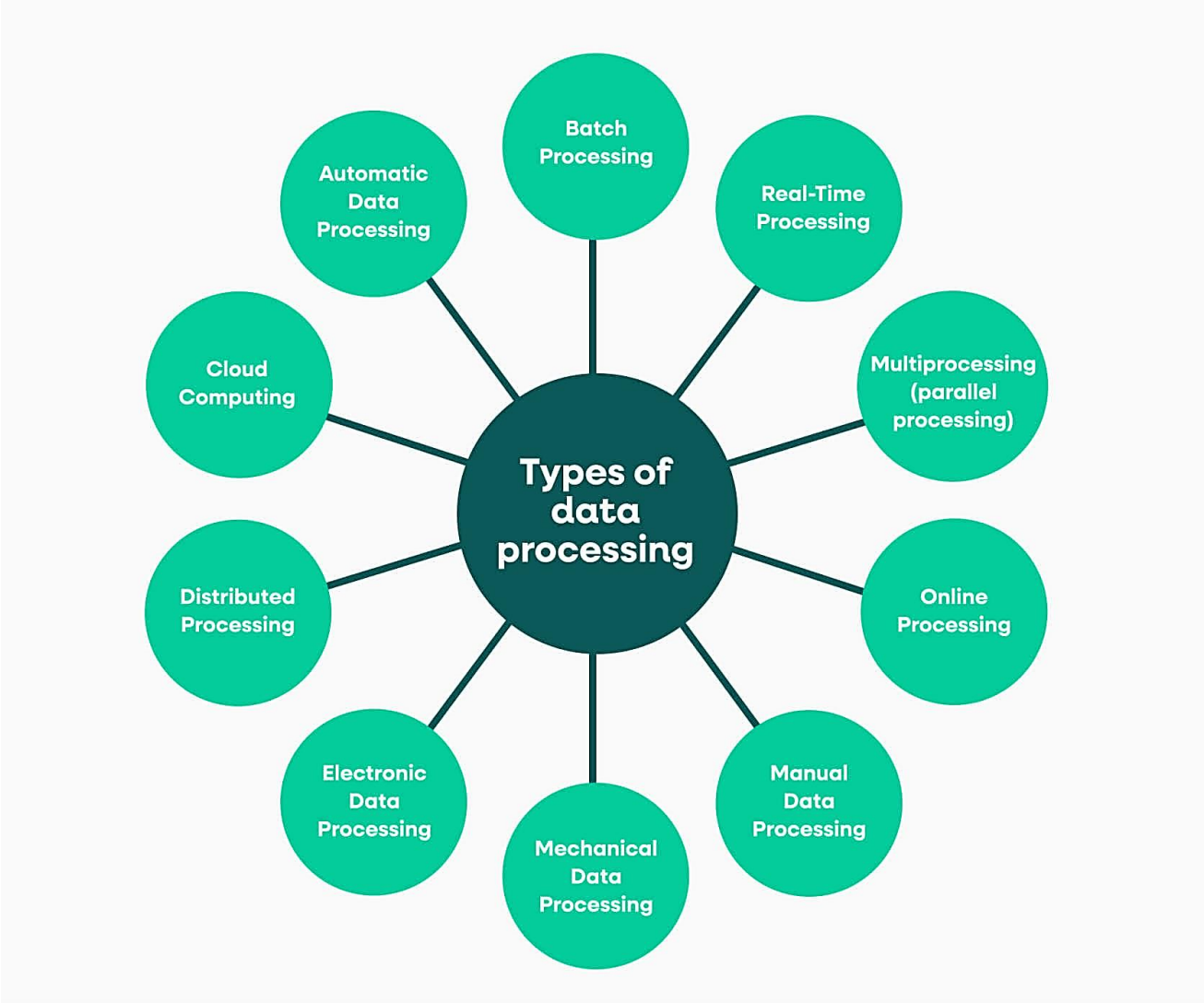
DATA PROCESSING WITH APACHE SPARK

The data processing cycle consists of a **series of steps where raw data (input) is fed into a system to produce actionable insights (output)**. Each step is taken in a specific order, but the entire process is repeated in a cyclic manner. The first data processing cycle's output can be stored and fed as the input for the next cycle



DATA PROCESSING WITH APACHE SPARK

Data processing utilizes various methods to convert raw data into meaningful information. These methods can be classified into several types, each catering to different scenarios and requirements.

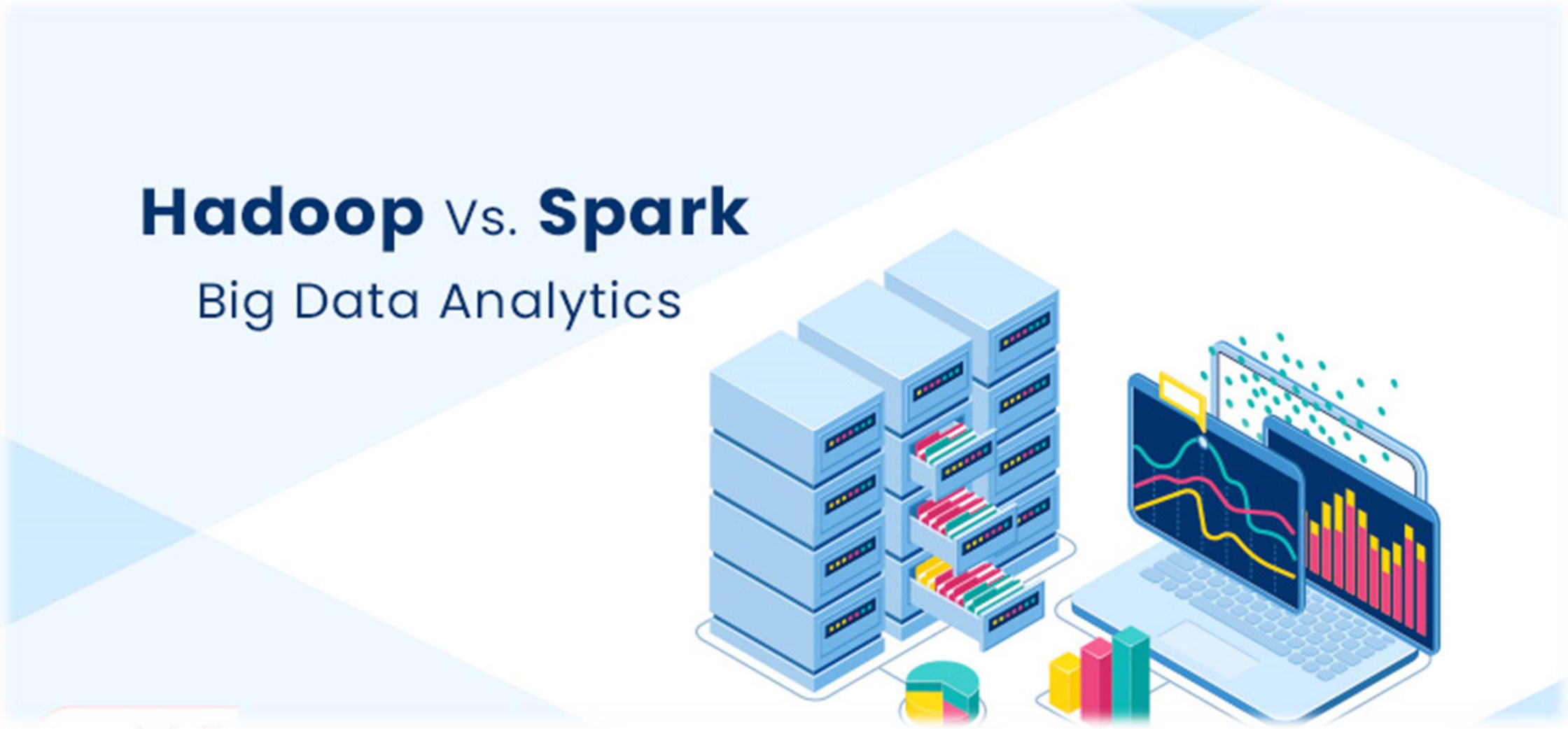


DATA PROCESSING WITH APACHE SPARK

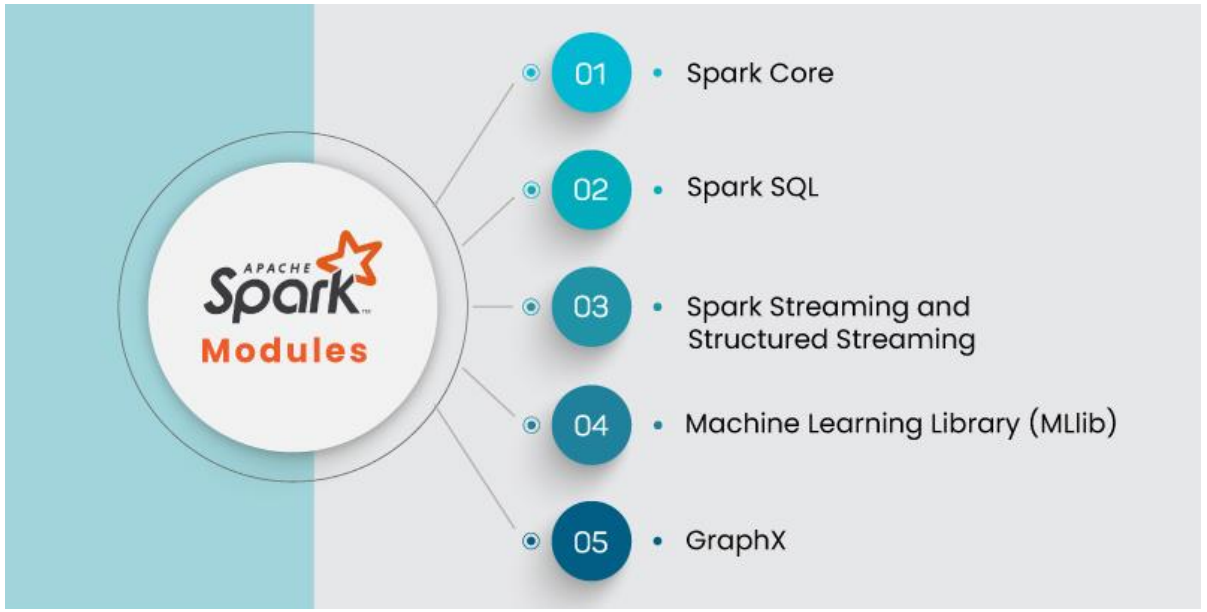
Apache Spark has been **one of the leading big data processing systems** on the market. The open-source platform has been proven to be the **preferred choice of enterprises for data processing, querying, and generating analytical reports**. Apache Spark is a fast and general-purpose cluster computing system. It provides **high-level APIs in Java, Scala, Python, and R** and an optimized engine that supports general execution graphs. Its **in-memory data processing abilities, along with adaptability and scalability**, make it a better choice than older big data processing models like Hadoop.

Apache Spark is **an open-source, lightning-fast computation technology** built based on **Hadoop and MapReduce technologies** that support various computational techniques for fast and efficient processing. Spark is known for its **in-memory cluster computation** which is the main contributing feature for increasing the processing speed of the spark applications.

DATA PROCESSING WITH APACHE SPARK



DATA PROCESSING WITH APACHE SPARK



DATA PROCESSING WITH APACHE SPARK

Apache Hadoop allows you to **cluster multiple computers to analyze massive dataset**s in parallel more quickly.

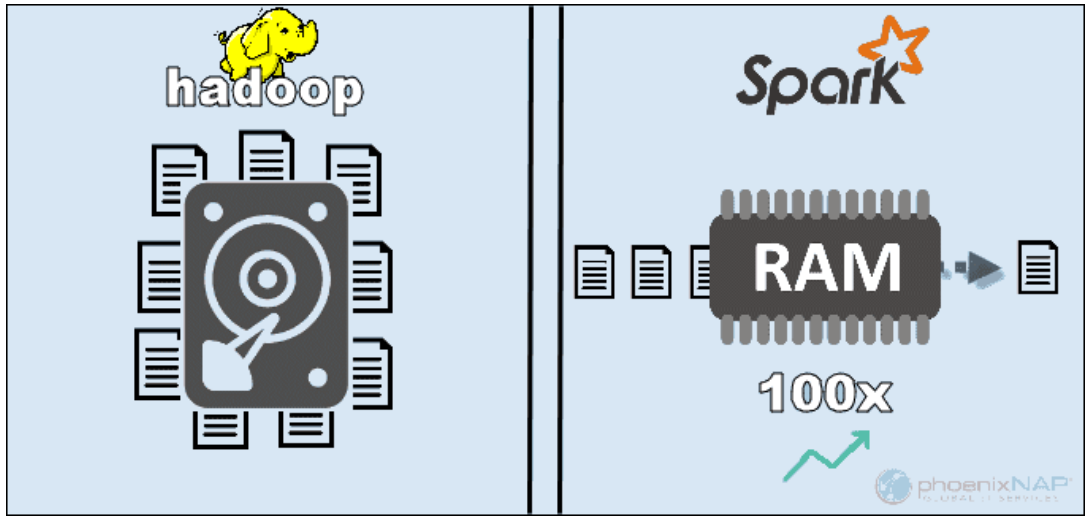
Apache Spark **uses in-memory caching and optimized query execution** for fast analytic queries against data of any size.

Hadoop stores and processes data on external storage. Spark stores and process data on internal memory. Hadoop processes data in **batches**. Spark processes data in **real time**.

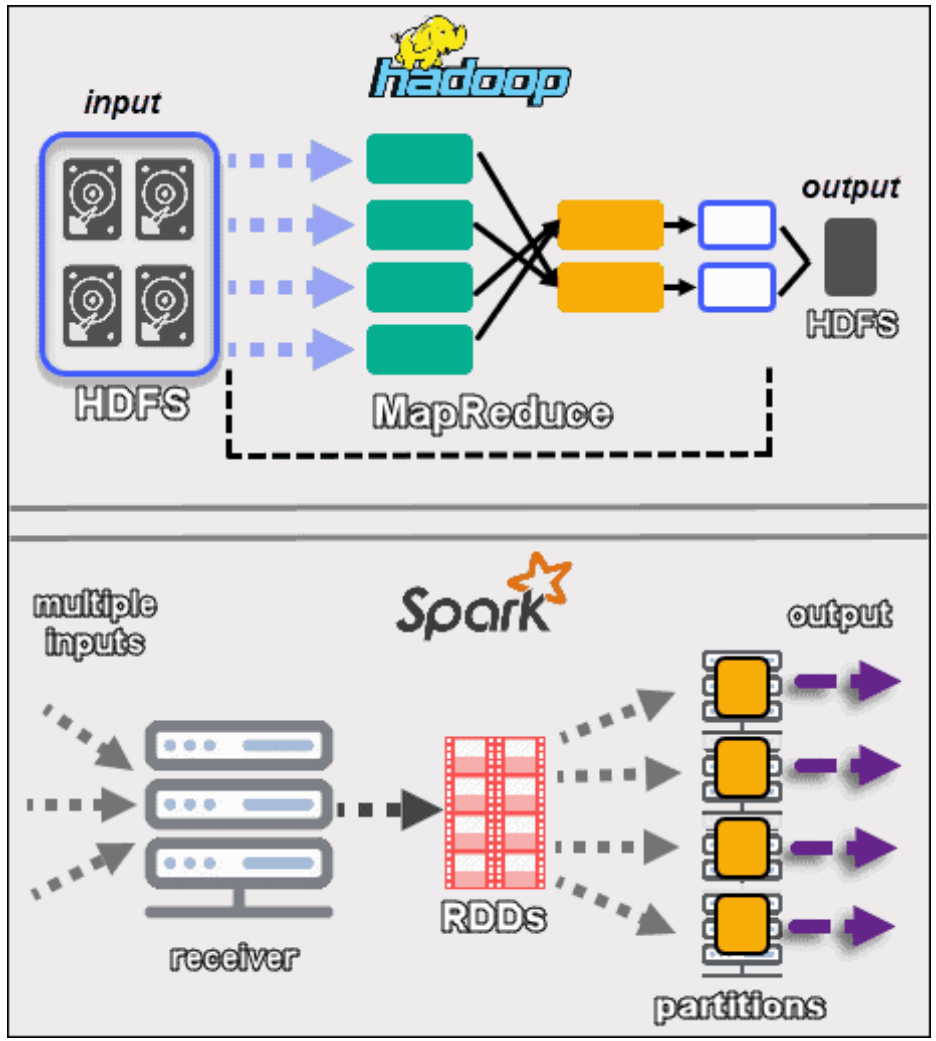
DATA PROCESSING WITH APACHE SPARK

IntelliPaat

WHAT IS	WHAT IS
hadoop	APACHE Spark
Big data processing engine	**Data Analytics Engine**
○ Hadoop Distributed File System (HDFS)	○ Spark Core
○ MapReduce Programming Model	○ Spark SQL
○ YARN	○ Spark Streaming

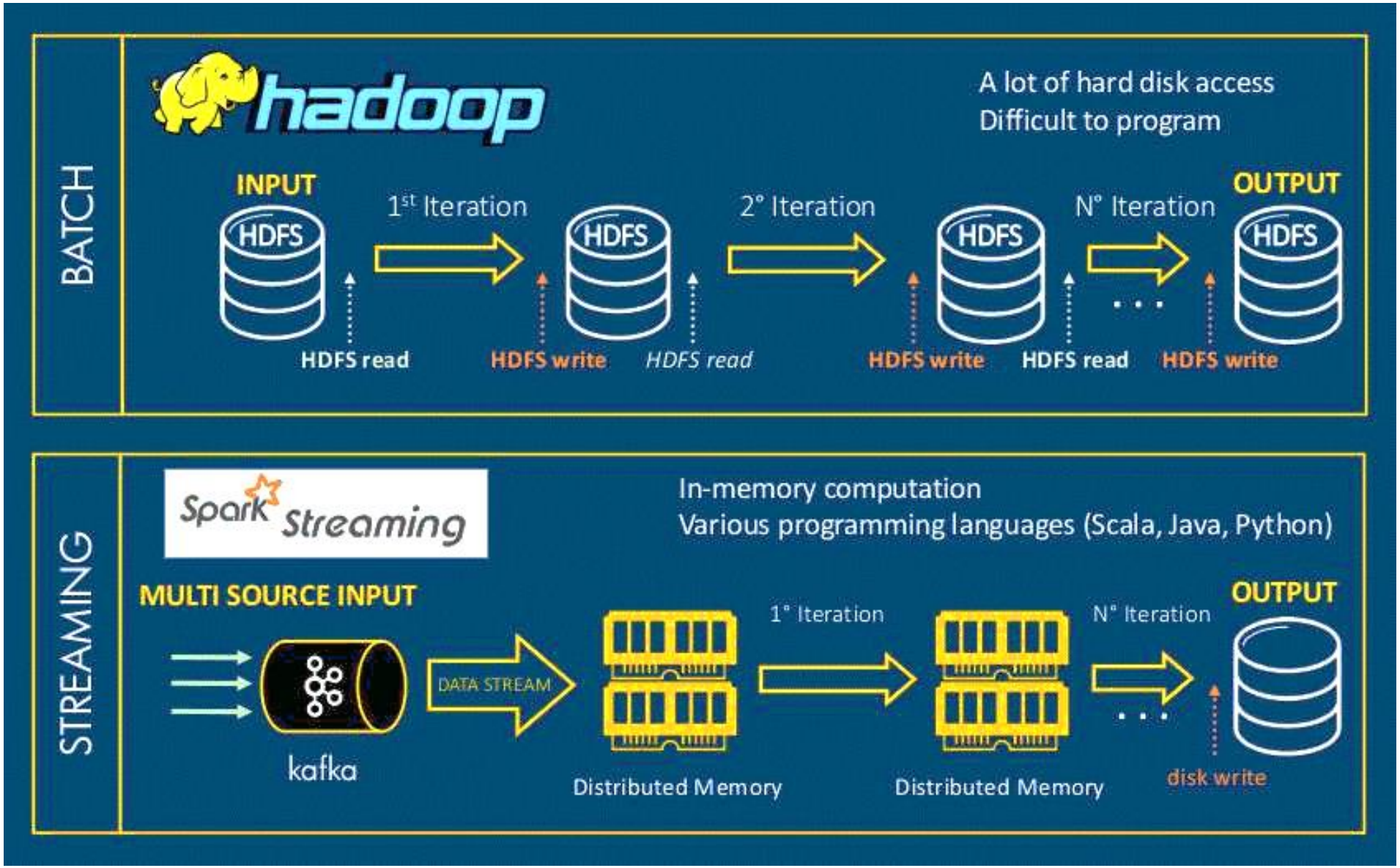


DATA PROCESSING WITH APACHE SPARK



Apache Spark works with **resilient distributed datasets (RDDs)**. An RDD is a **distributed set of elements stored in partitions on nodes across the cluster**. The size of an RDD is usually too large for one node to handle. Therefore, Spark partitions the RDDs to the closest nodes and performs the operations in parallel. The **system tracks all actions performed on an RDD by the use of a Directed Acyclic Graph (DAG)**.

DATA PROCESSING WITH APACHE SPARK



Use Cases of Hadoop vs Spark

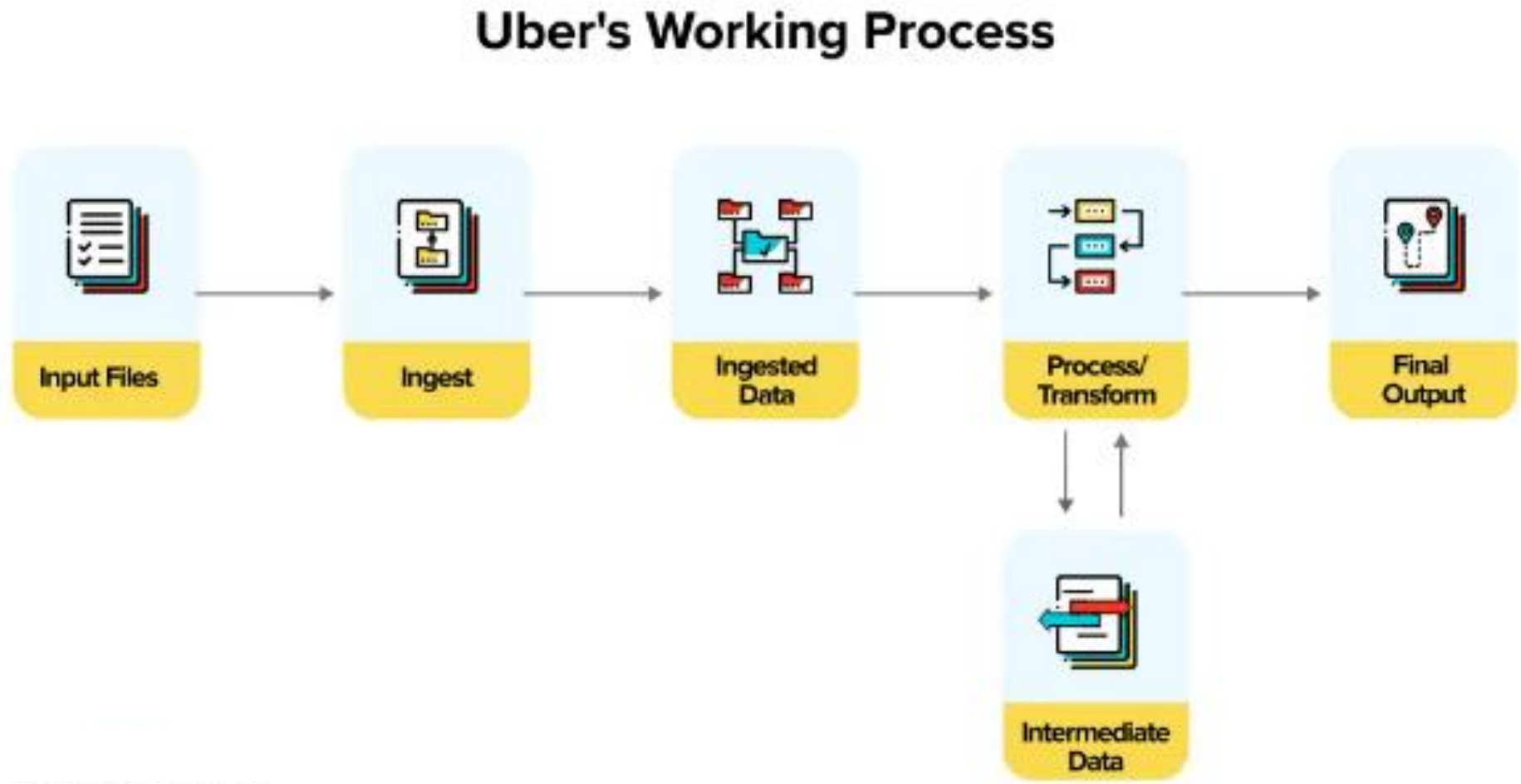
Hadoop use cases include:

1. Processing large datasets in environments where data size exceeds available memory.
2. Building data analysis infrastructure with a limited budget.
3. Completing jobs where immediate results are not required, and time is not a limiting factor.
4. Batch processing with tasks exploiting disk read and write operations.
5. Historical and archive data analysis.

Spark use cases include:

1. The analysis of real-time stream data.
2. When time is of the essence, Spark delivers quick results with in-memory computations.
3. Dealing with the chains of parallel operations using iterative algorithms.

DATA PROCESSING WITH APACHE SPARK



DATA PROCESSING WITH APACHE SPARK

IntelliPaat

Hadoop vs Spark: Comparison Table		
Parameter	Hadoop	Spark
1. Intent	Data Processing Engine	Data Analytics Engine
2. Work Process	Analyses batches of data present in huge volumes	Analyses and processes real-time data
3. Processing Style	Batch mode	real-time data handling
4. Latency	High	Low
5. Scalability	It only requires the addition of nodes and disks, which makes it quickly scalable	Complex scalability due to more reliability on RAM
6. Cost	Less costly due to the MapReduce Model	Is more expensive due to its in-memory solution
7. Security	More secure	Less Secure
8. Ease of Use	Complex to use	Easier to use

Which one is Better: Hadoop or Spark?

While **Spark is faster than thunder** and is easy to use, **Hadoop comes with robust security, mammoth storage capacity, and low-cost batch processing capabilities**. Choosing one out of two depends entirely upon your project's requirement, the other alternative being combining parts of Hadoop and Spark to give birth to an unbeatable combination.

You can also consider using both Hadoop and Spark and get the best of two worlds experience and can call this new framework Spoop.

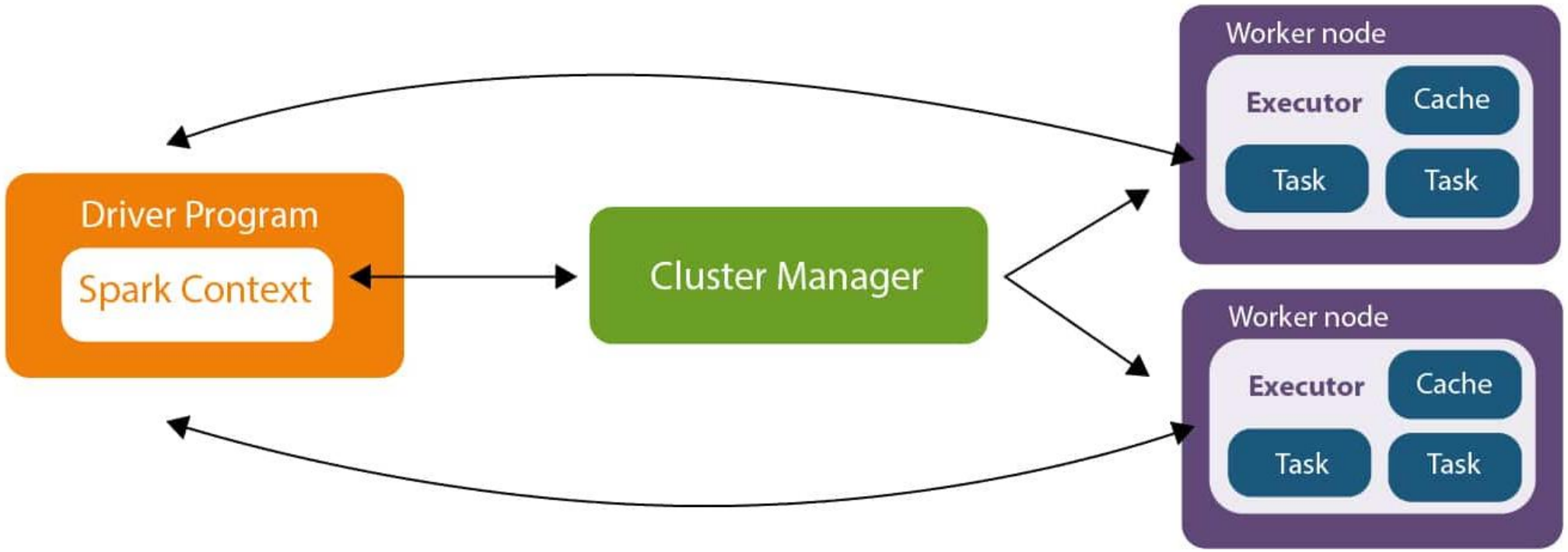
DATA PROCESSING WITH APACHE SPARK

Build ETL Pipelines in Apache Spark

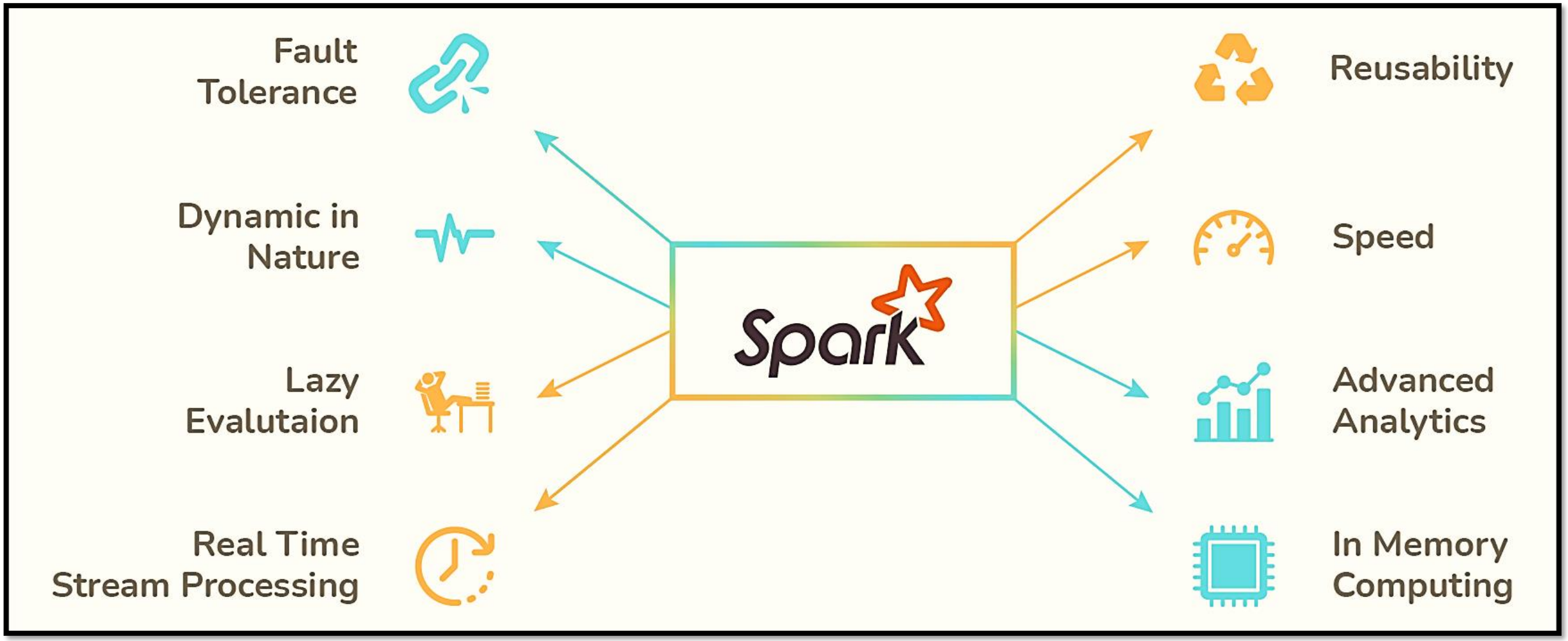
- The reason is Apache Spark being "Fast and general engine for large-scale data processing". Spark is a fast and general processing engine compatible with Hadoop data.
- This also provides in memory compute with language support for Scala, R, Python and SQL.
- Data transformation/engineering can be done in notebooks with statements in different languages.
- Able to run each step of the process in a notebook, so step by step debugging is easy. We will also be able to see this process during job execution, so it is easy to see if your job stops.
- Clusters can be configured in a variety of ways, both regarding the number and type of compute nodes.
- Flexibility in Coding , the programmatic approach provides the flexibility of fine-tuning codes to optimize performance.

Apache Spark has 24.1K GitHub stars and 20.4K forks on GitHub

Spark Cluster Architecture



APACHE SPARK FEATURES



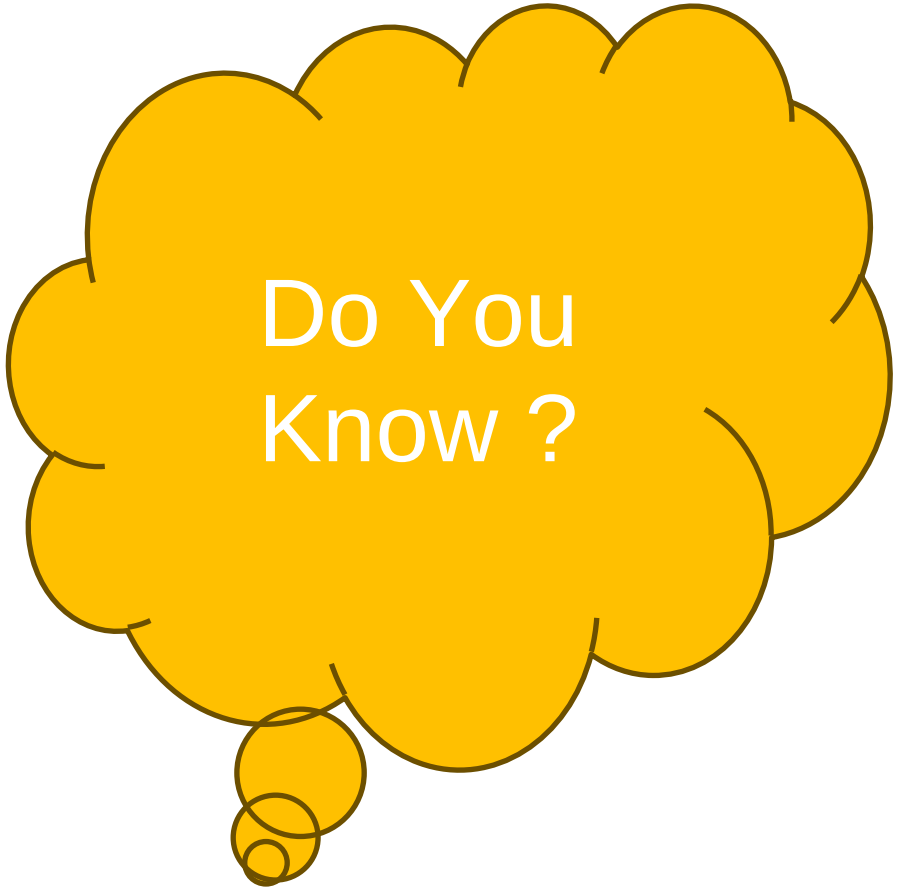
APACHE SPARK FEATURES

RDD stands for **Resilient Distribution Datasets**. It is a fault-tolerant collection of parallel running operational elements. The partitioned data of RDD is distributed and immutable.

There are two types of datasets:

Parallelized collections: Meant for running parallelly.

Hadoop datasets: These perform operations on file record systems on HDFS or other storage systems



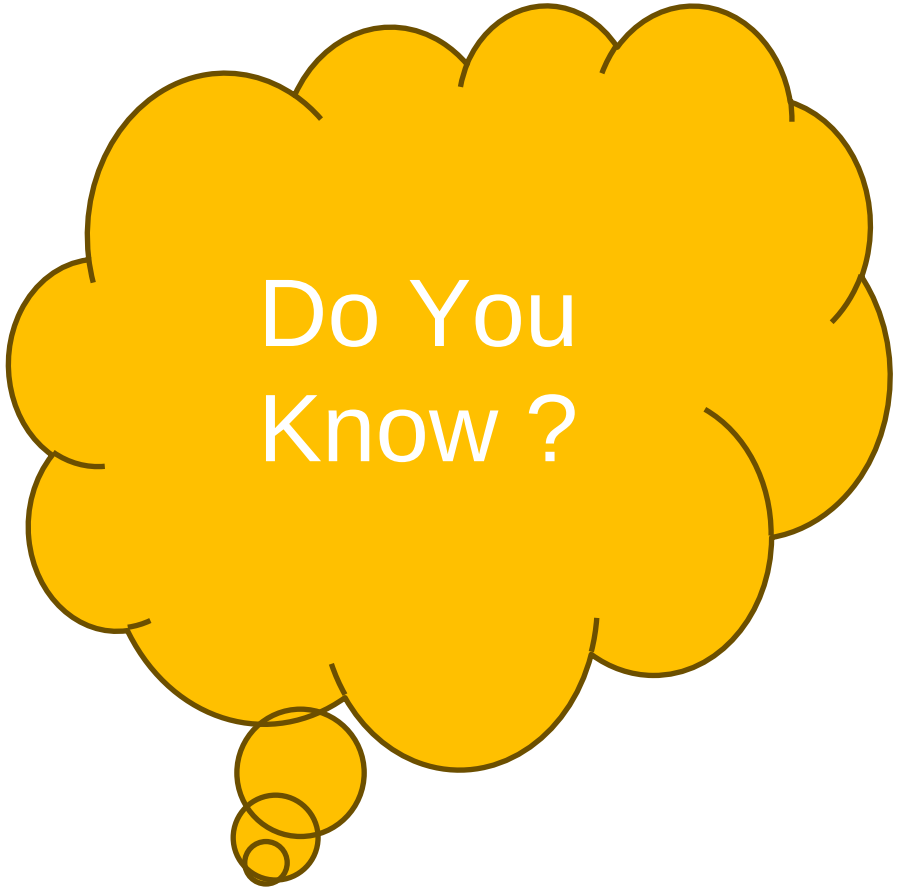
APACHE SPARK FEATURES

The most widely-used engine for scalable computing

Thousands of companies, including 80% of the Fortune 500, use Apache Spark™. Over 2,000 contributors to the open source project from industry and academia.

Ecosystem

Apache Spark™ integrates with your favorite frameworks, helping to scale them to thousands of machines.



APACHE SPARK ECO SYSTEM

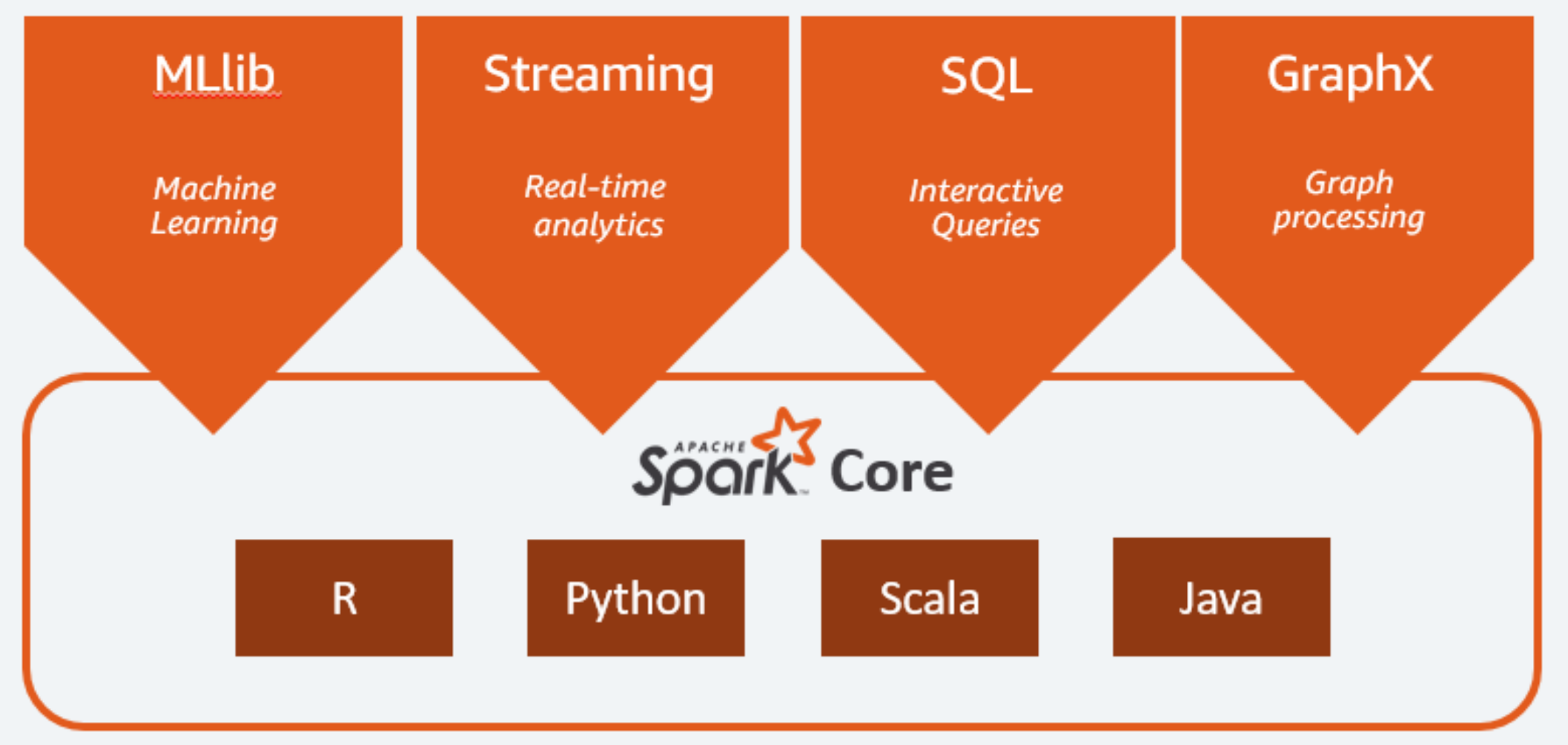
IntelliPaat

Data science and Machine learning

SQL analytics and BI

Storage and Infrastructure

APACHE SPARK WORKLOADS



Apache Flink

Contact Us

080-4524-9465

support@intellipaat.com