SPARK

Surendra Panpaliya

GKTCS Innovations

https://www.gktcs.com

Agenda

- Spark Basics
- Spark Architecture
- Working with RDD's
- RDD Operations
- Passing Functions to spark

Spark Basics

Surendra Panpaliya

GKTCS Innovations

https://www.gktcs.com

Spark Basics

What is Spark?

History of Apache Spark

Features of Apache Spark

Uses of Spark

Spark Installation

What is Spark?

An Open-source,

General-purpose

Lightning fast

Cluster computing framework.

What is Spark?

Built on the top of the Hadoop MapReduce.

Optimized to run in Memory

Increases the processing speed of an application.

Hadoop's MapReduce writes data

to and from computer hard drives.

History of Apache Spark

Initiated by Matei Zaharia

at UC Berkeley's AMPLab in 2009.

open sourced in 2010

In 2013, the project was acquired by

Apache Software Foundation.

History of Apache Spark

Feb 2014 onwords

the Spark emerged as a

Top-Level Apache Project

Features of Apache Spark

- Fast
- Easy to Use
- Generality
- Lightweight
- Runs Everywhere

Fast

It provides high performance

for both Batch and Streaming data,

Using a DAG scheduler,

a query optimizer, and

a physical execution engine.

Easy to Use

It facilitates

to write the application in

Java, Scala, Python, R, and SQL.

It also provides more than 80 high-level operators.

Generality

It provides a collection of libraries

SQL and DataFrames,

MLlib for machine learning,

GraphX,

Spark Streaming

Lightweight

It is a light unified analytics engine

which is used

for large scale

data processing.

Runs Everywhere

It can easily run on Hadoop,

Apache Mesos,

Kubernetes,

standalone, or

in the cloud.

Uses of Spark

Surendra Panpaliya

GKTCS Innovations

https://www.gktcs.com

Uses of Spark

Data integration

Stream processing

Machine learning

Interactive analytics

Data integration

The data generated by systems

are not consistent enough

to combine for analysis.

Data integration

To fetch consistent data

from systems

Use processes like

Extract, Transform, and Load (ETL).

Handle the real-time generated data

Such as log files.

Operate streams of data

Refuses potentially fraudulent operations.

Stream processing

Machine learning

Machine learning approaches

become more feasible and

increasingly accurate

due to enhancement in the volume of data.

Interactive analytics

Spark is able

to generate the respond rapidly.

Instead of running pre-defined queries,

Can handle the data interactively.

Spark Installation

- Download Apache Spark
- https://spark.apache.org/downloads.html
- https://www.apache.org/dyn/closer.lua/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
- tar -xvzf spark-3.2.1-bin-hadoop3.2.tgz

Spark Installation

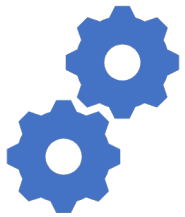
- vim .zshrc # Path Environment Set up for Mac
- SPARK_HOME= /Users/surendra/Spark3_2/spark-3.2.1-bin-hadoop3.2
- export PATH=$SPARK_HOME/bin:$PATH
- source .zshrc
- Let's test the installation on the command prompt type
- spark-shell

Spark Architecture

Surendra Panpaliya

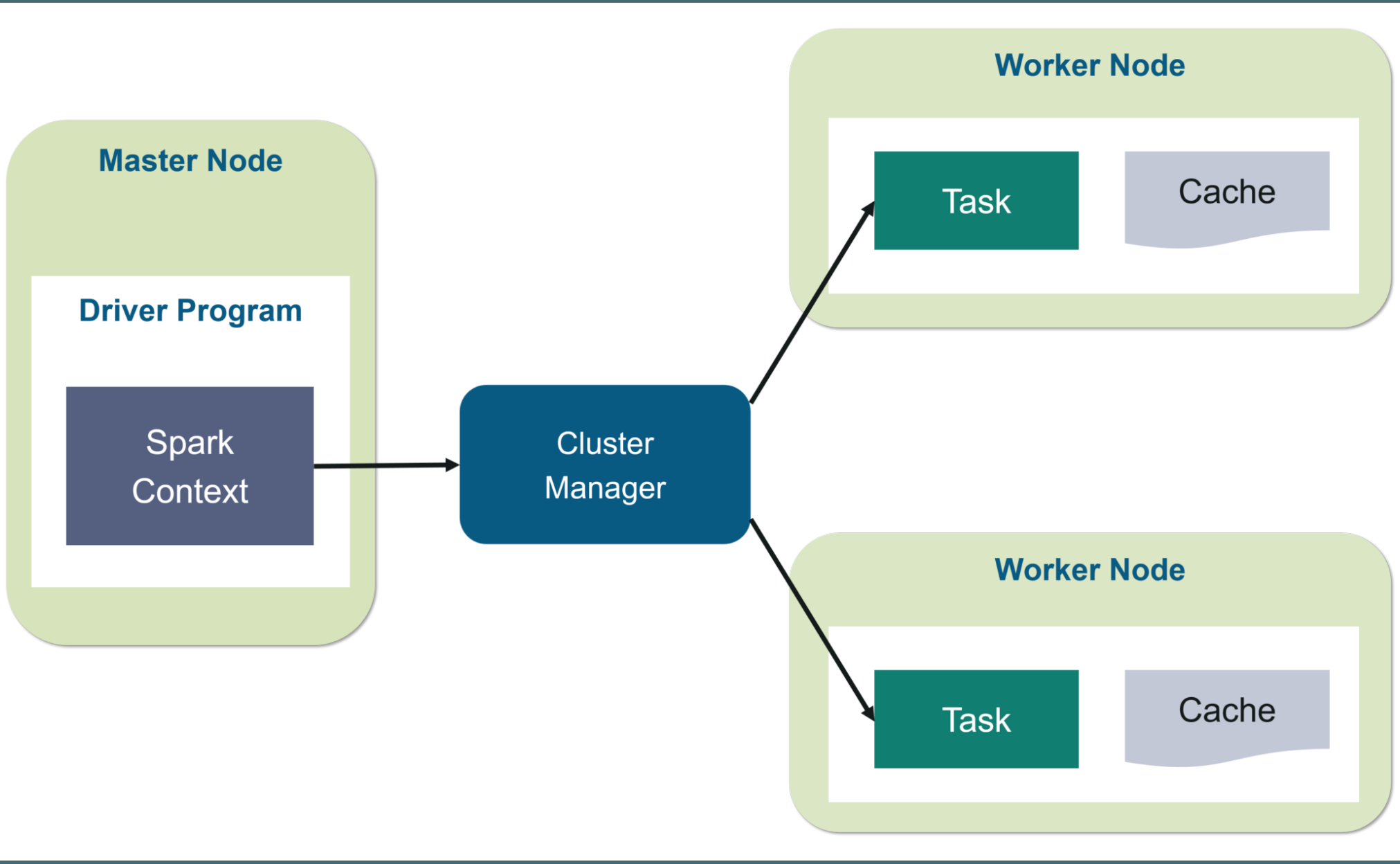
GKTCS Innovations

https://www.gktcs.com



Spark Architecture

- Master-slave architecture.
- Cluster consists of
- Single master & Multiple slaves.


Worker Node
Task
Cache
Master Node
Driver Program
Spark Context
Cluster Manager
Worker Node
Task
Cache

Spark Architecture

The Spark architecture depends upon:

Resilient Distributed Dataset (RDD)

Directed Acyclic Graph (DAG)

Resilient Distributed Datasets (RDD)

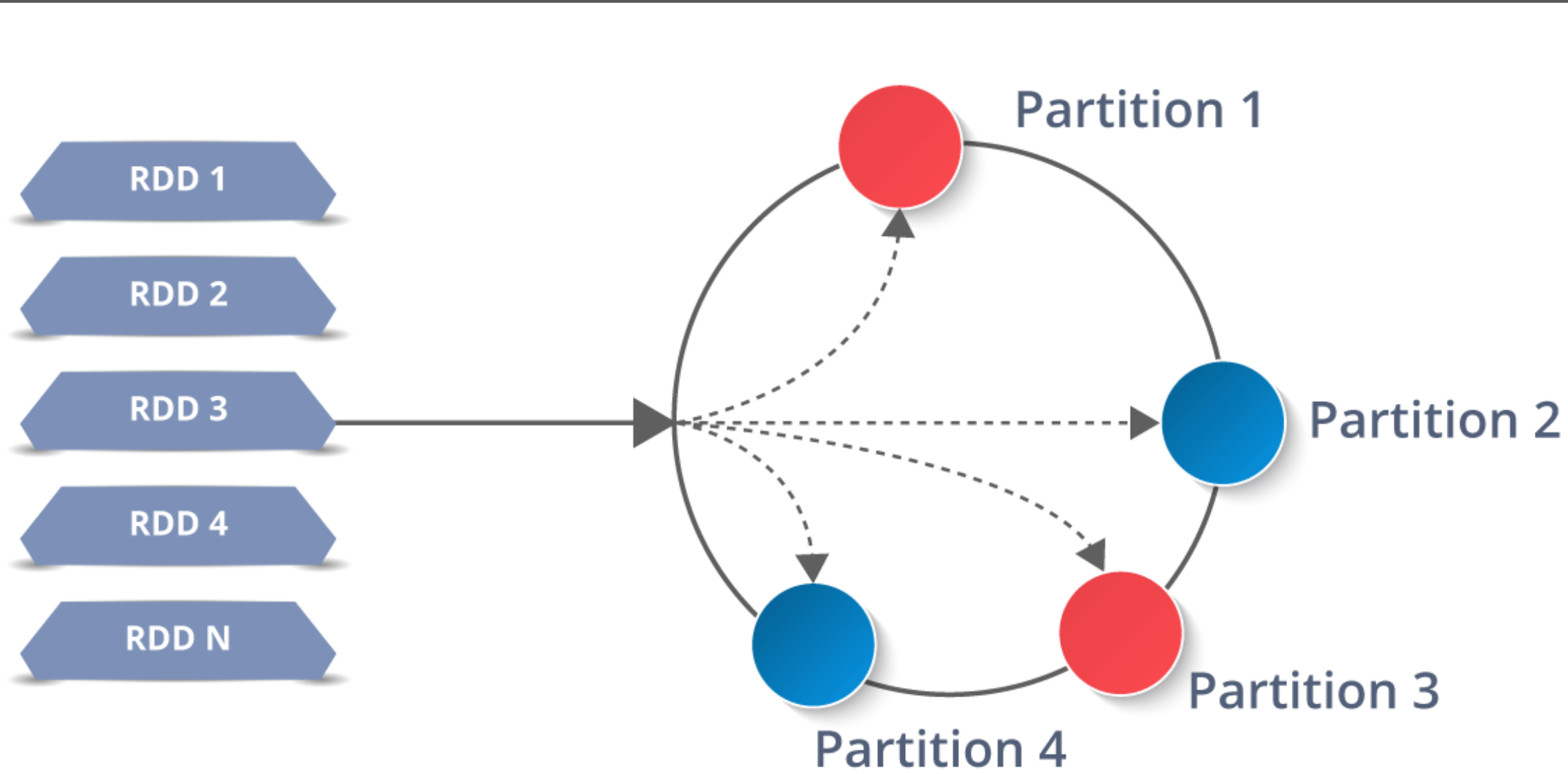
Group of data items

Can be stored in-memory on worker nodes.

Resilient: Restore the data on failure.

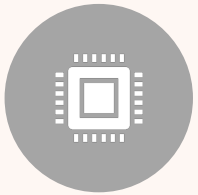
Distributed: Data is distributed among different nodes.

Dataset: Group of data.

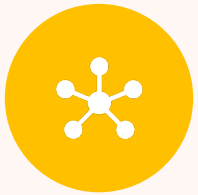

RDD 1
RDD 2
RDD 3
RDD 4
RDD N
Partition 1
Partition 2
Partition 3
Partition 4

Directed Acyclic Graph (DAG)

Finite direct graph



Performs a sequence of computations on data.

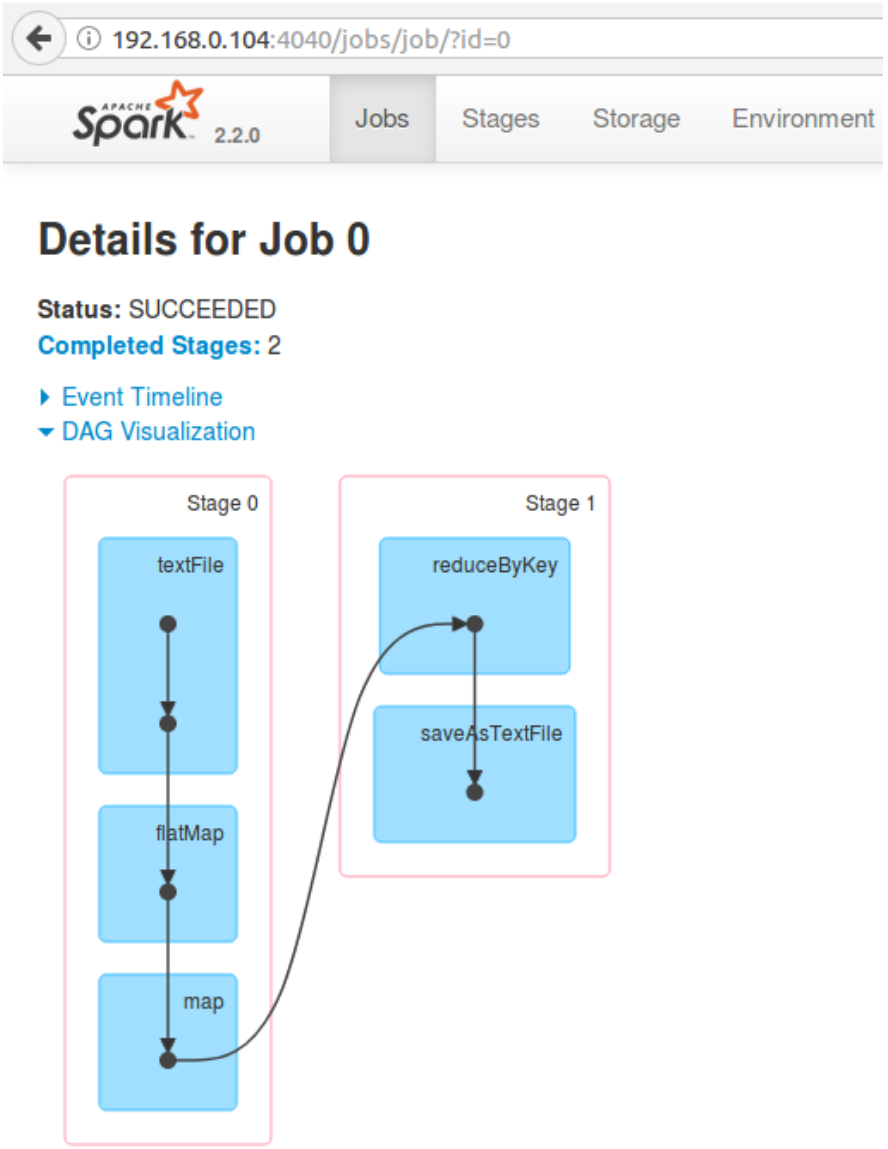


Each node is an RDD partition

Edge is a transformation on top of data.

Graph -> Navigation

Directed and Acyclic -> How it is done?


192.168.0.104:4040/jobs/job/?id=0
Spark 2.2.0
Jobs
Stages
Storage
Environment
Details for Job 0
Status: SUCCEEDED
Completed Stages: 2
Event Timeline
DAG Visualization
Stage 0
textFile
flatMap
map
Stage 1
reduceByKey
saveAsTextFile

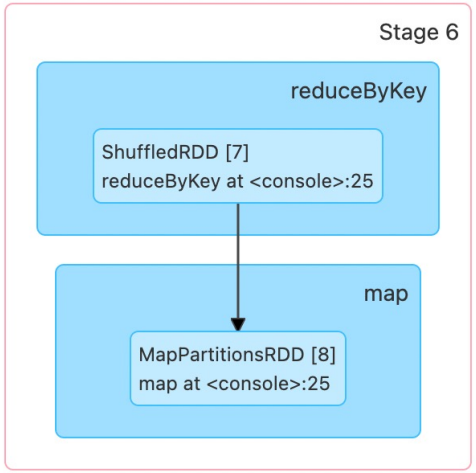
Not Secure | http://192.168.0.102:4040/stages/stage/?id=6&attempt=0

Spark 3.1.2 Jobs Stages Storage Environment Executors

Details for Stage 6 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 0.2 s
Locality Level Summary: Node local: 2
Shuffle Read Size / Records: 558.0 B / 46
Shuffle Write Size / Records: 275.0 B / 23
Associated Job Ids: 3

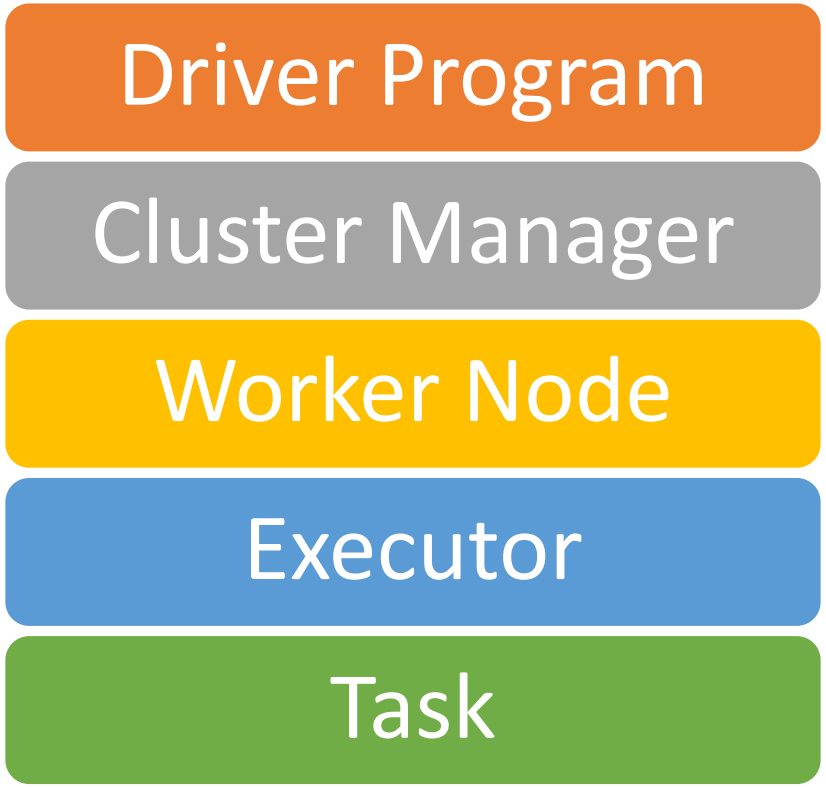
▼ DAG Visualization



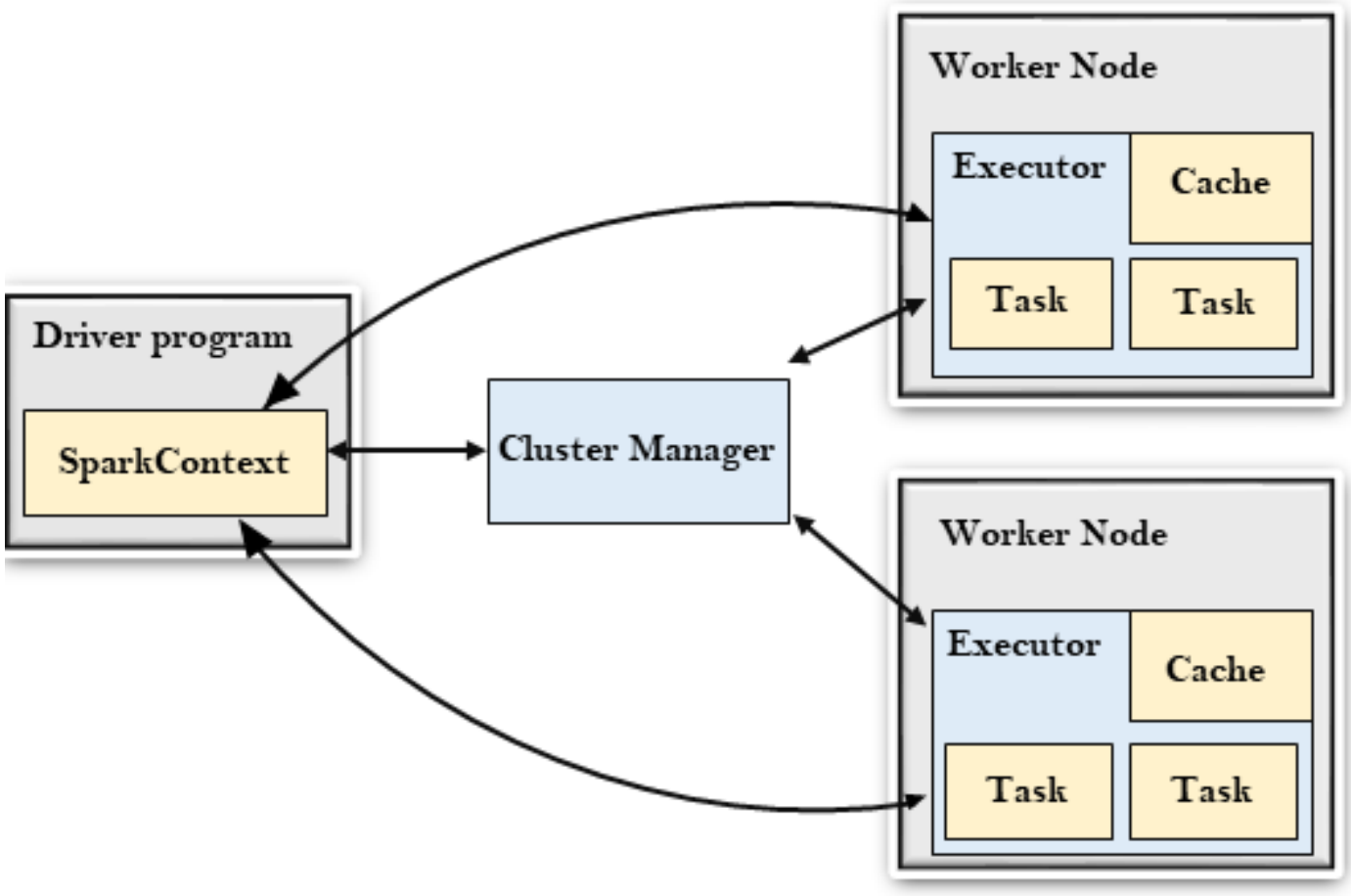
▶ Show Additional Metrics
▶ Event Timeline

Summary Metrics for 2 Completed Tasks

Spark Architecture



Spark Architecture



Driver Program

Process that runs

the main() function

of the application and

creates the **SparkContext** object.

SparkContext

The purpose of **SparkContext** is

to coordinate the spark applications,

running as independent

sets of processes on a cluster.

SparkContext

To run on a cluster,

the **SparkContext** connects

to a different type of

cluster managers

SparkContext

Perform the Following Tasks: -

1. It Acquires Executors

on nodes

in the cluster.

SparkContext

2. Sends your application code

to the executors.

3. Sends Tasks to

the Executors to run.

Cluster Manager

The role of the cluster manager is

to **allocate resources** across applications.

The Spark is capable enough of

running on a large number of clusters.

Cluster Manager

Consists of various types of cluster managers

Hadoop YARN,

Apache Mesos

Standalone Scheduler.

Cluster Manager

Standalone Scheduler

is a standalone spark cluster manager

that facilitates

to install Spark on

an empty set of machines.

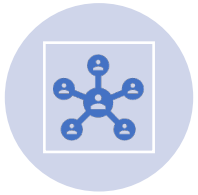
Worker Node

- Is a slave node
- Role is to run
- the application code
- in the cluster.

Executor

An executor is a process launched

for an application



on a worker node.

It runs tasks

keeps data in memory or

disk storage across them.

Executor

It read and write data

to the external sources.

Every application contains

its executor.

Task

A unit of work

that will be sent

to one executor.

Components of Spark

Apache Spark Core

Spark SQL

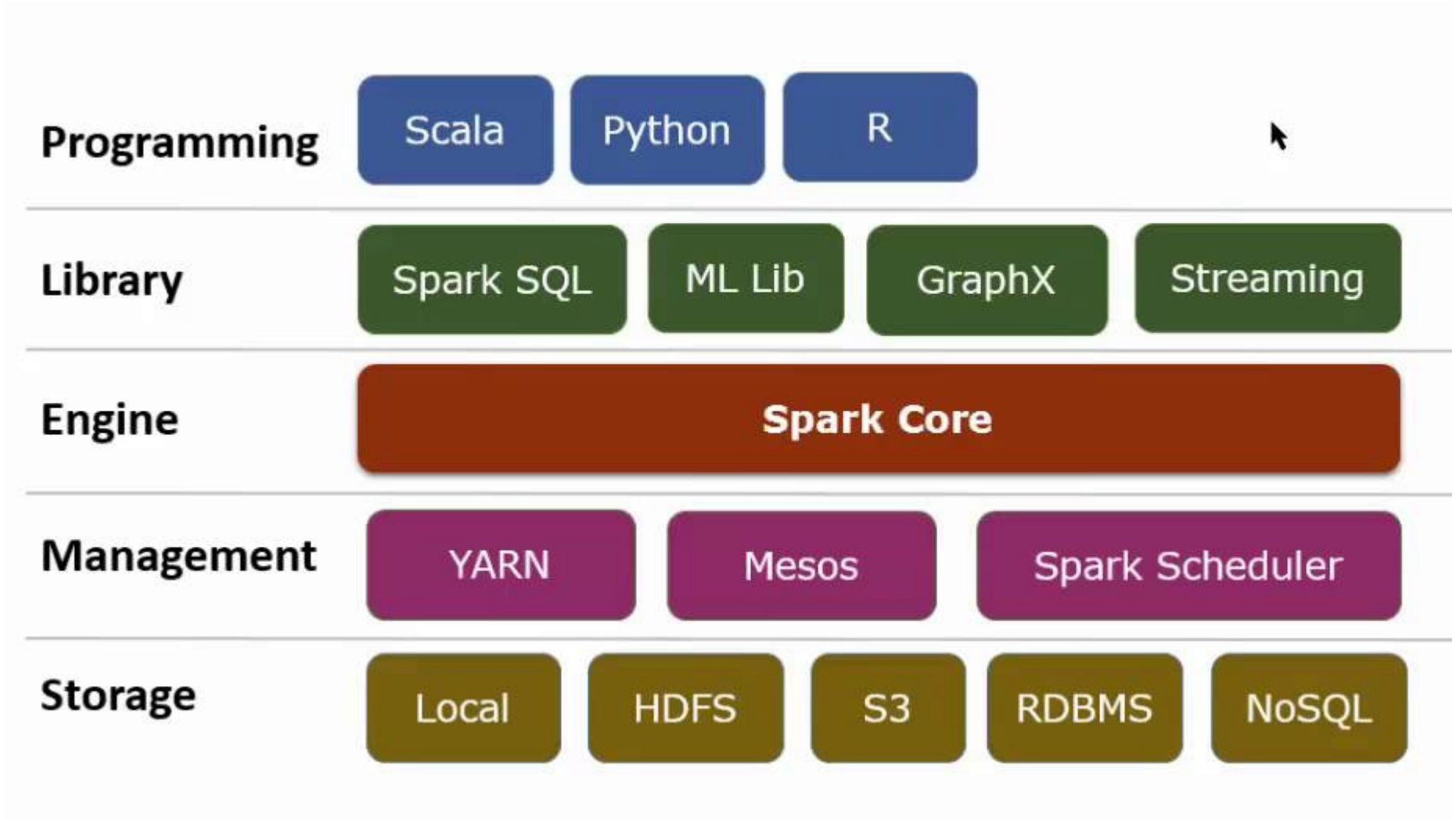
Spark Streaming

MLlib (Machine Learning Library)

GraphX

SparkR


Spark SQL
Spark streaming
Spark MLlib
Spark GraphX
Spark R
Spark
Scala
R


Programming
Scala
Python
R
Library
Spark SQL
ML Lib
GraphX
Streaming
Engine
Spark Core
Management
YARN
Mesos
Spark Scheduler
Storage
Local
HDFS
S3
RDBMS
NoSQL

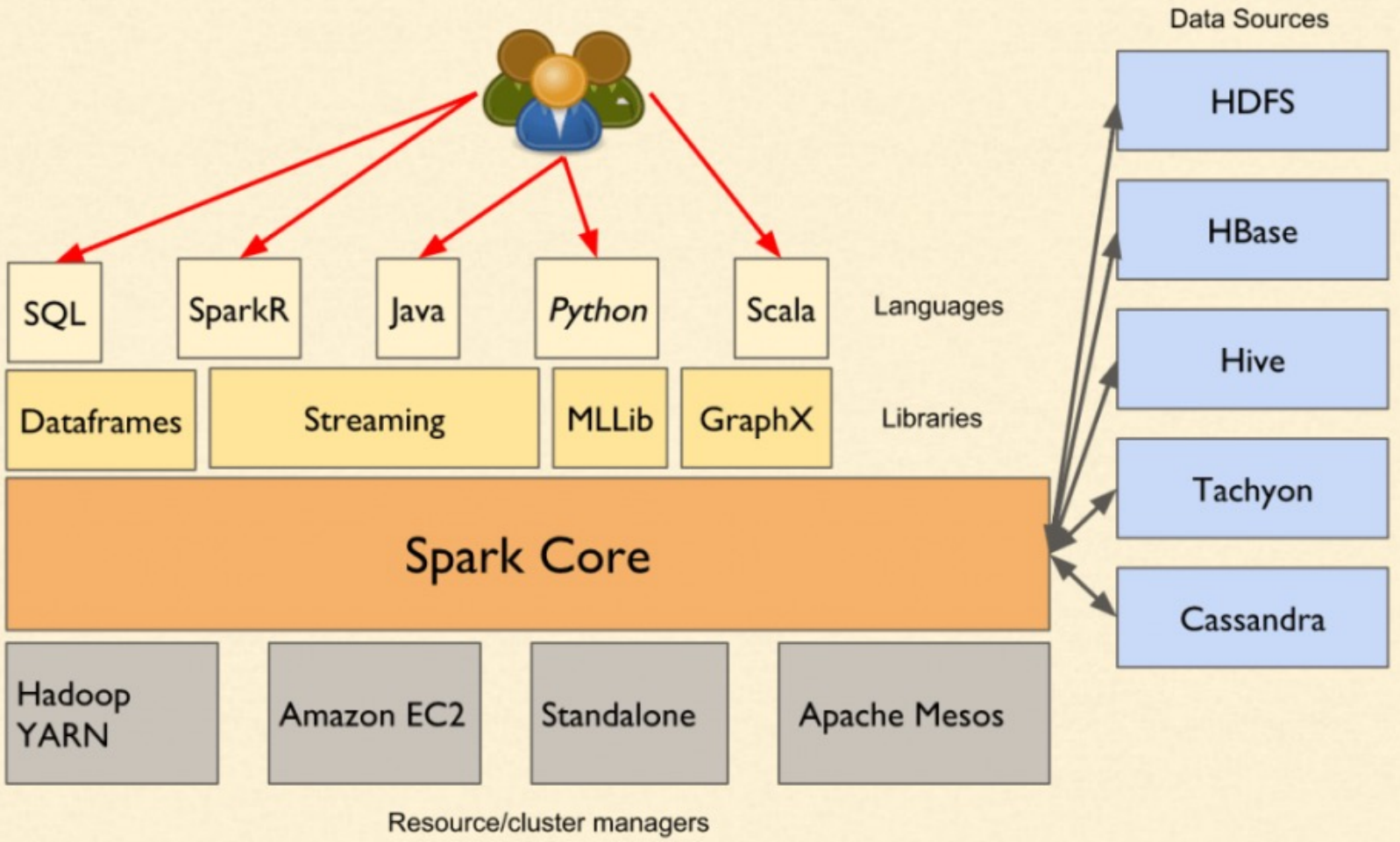
Apache Spark Core

- Heart of Spark
- Performs the core functionality.
- Written on Scala
- which runs on top of JVM
- Supports languages like Java, Python, R

Apache Spark Core

- It holds the components for
- Task scheduling,
- Fault recovery,
- Interacting with storage systems and
- Memory management.

Spark Architecture



Spark SQL

The Spark SQL is built on the top of Spark Core.

It provides support for structured data.

Allows to query the data via

SQL & HQL (Hive Query Language)

Spark Streaming

supports scalable and fault-tolerant

processing of streaming data.

Uses Spark Core’s

fast scheduling capability

to perform

streaming analytics.

Spark SQL

Supports JDBC and ODBC connections

that establish a relation between

Java objects and existing databases,

data warehouses and

business intelligence tools.

Spark SQL

Supports various sources of data like

Hive tables, Parquet, and JSON.

Supports structured and semi-structured data

Query data for CSV, JSON,

Sequence and Parquet file formats

Spark Streaming

It accepts data in

mini-batches and

performs RDD transformations

on that data.

Used to analyze batches of historical data

Spark Streaming

The log files

generated by

web servers

can be considered as

a real-time

example of a data stream

MLlib

- Contains Machine learning algorithms.
- Include correlations and hypothesis testing,
- Classification, Regression,
- Clustering
- principal component analysis.

MLlib

It is nine times faster than

the disk-based implementation

used by Apache Mahout.

GraphX

Used to

manipulate graphs

perform

graph-parallel computations.

GraphX

It facilitates

to create a directed graph

with arbitrary properties

attached to

each vertex and edge.

GraphX

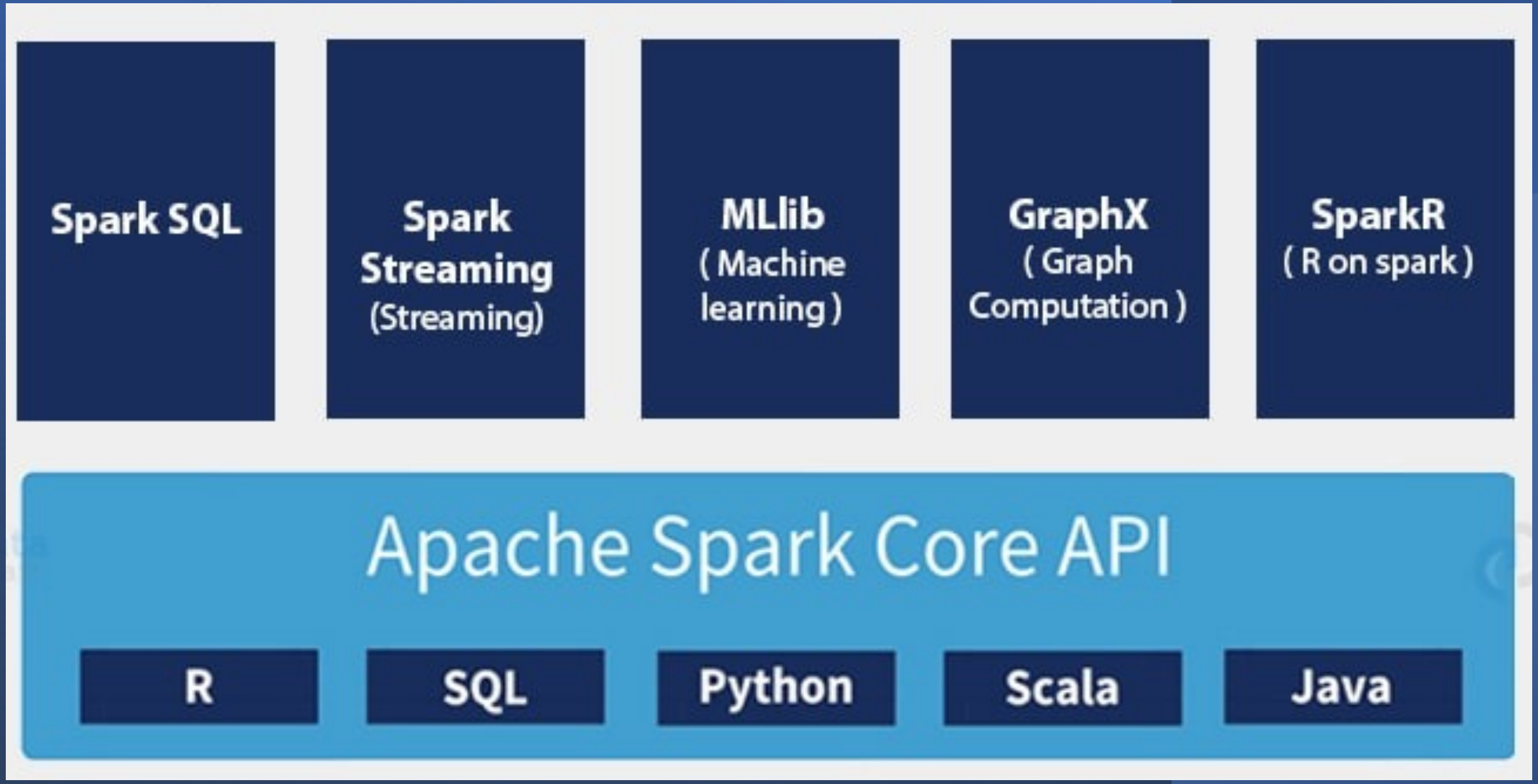
To manipulate graph,

it supports various

fundamental operators

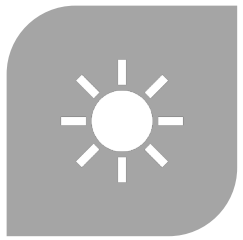
like subgraph, join Vertices, and

aggregate Messages.

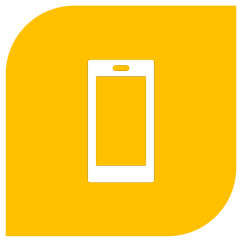

Spark SQL
Spark Streaming (Streaming)
MLlib (Machine learning)
GraphX (Graph Computation)
SparkR (R on spark)
Apache Spark Core API
R
SQL
Python
Scala
Java

SparkR

R PACKAGE



GIVES LIGHT-WEIGHT FRONTEND



TO USE

APACHE SPARK

FROM R.

SparkR

Allows data scientists

to analyze large datasets

interactively run jobs on them

from the R shell.

SparkR

The main idea behind SparkR was

to explore different techniques

to integrate

the usability of R

with the scalability of Spark.

Thank You

Surendra Panpaliya

GKTCS Innovations

https://www.gktcs.com