



*Spark* 

A

to

Z

DEEPA  
VASANTHKUMAR

V2.

# A

<b>Action</b>	Operations that trigger the execution of the RDD transformations and return a result to the driver program or write it to storage.
<b>API</b>	A set of functions and protocols for building software and applications, allowing interaction with Apache Spark.
<b>Apache Spark</b>	An open-source distributed computing system for big data processing and analytics.

DEEPA

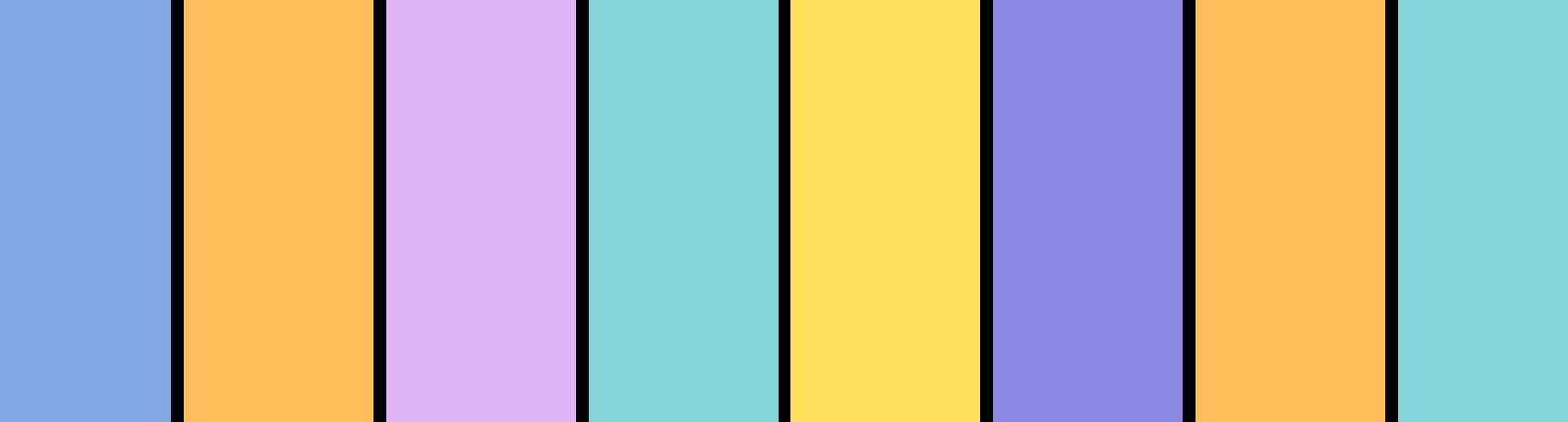
VASANTHKUMAR

# A

<b>Adaptive Query Execution (AQE)</b>	A feature in Spark that dynamically adjusts query plans at runtime based on the actual data being processed, leading to optimized performance.
<b>Aggregation</b>	A process of combining multiple values into a single value. Spark provides various aggregation functions such as <code>sum()</code> , <code>avg()</code> , <code>count()</code> , and custom aggregations using <code>aggregate()</code> .

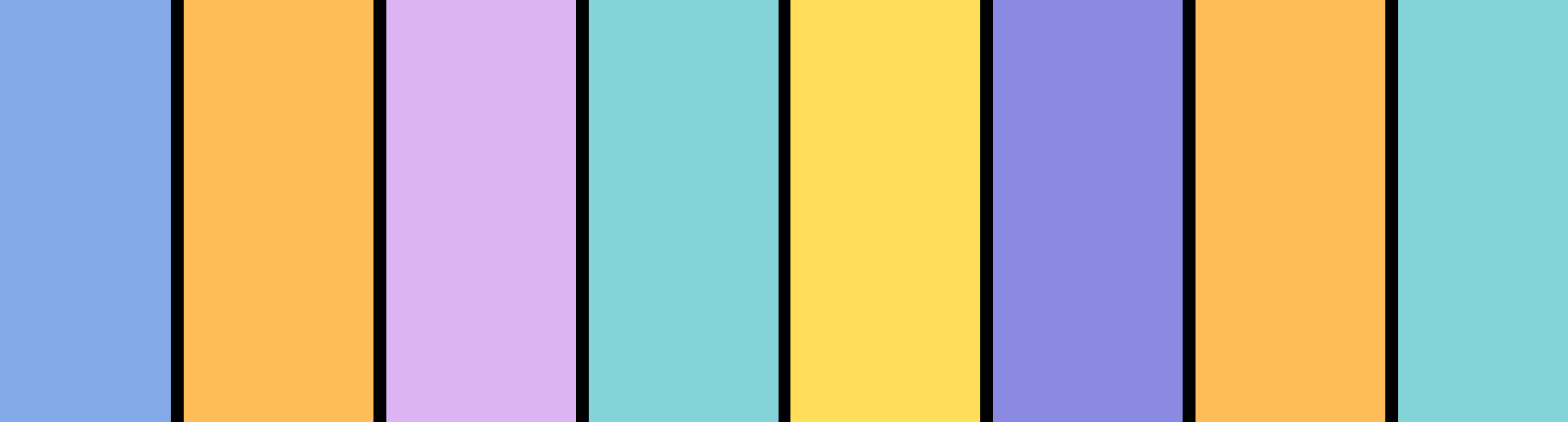
DEEPA

VASANTHKUMAR



# B

<b>Broadcast variables</b>	Variables cached on each machine, instead of shipping a copy of it with tasks, to improve performance when tasks across stages need the same data.
<b>Batch Processing</b>	Processing of large blocks of data at once, as opposed to real-time or streaming data processing. Spark is often used for batch processing in ETL workflows.
<b>Bucketing</b>	A technique for partitioning data into a fixed number of buckets to optimize join operations and aggregation tasks.



# B

## Block

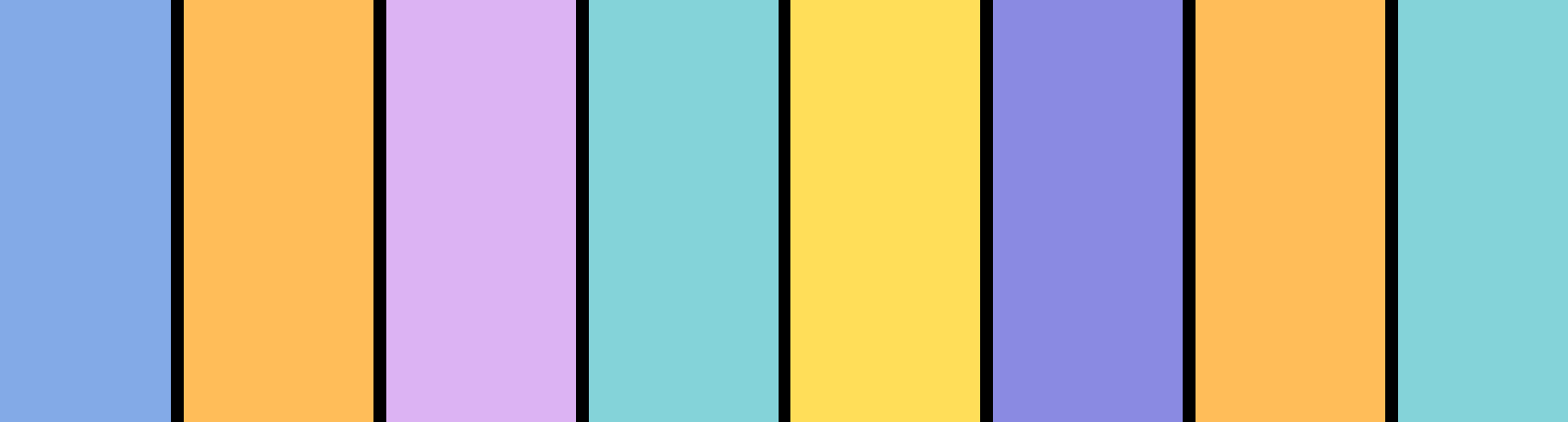
In Spark, a block is a unit of data storage. Blocks are pieces of data that are split and distributed across the cluster nodes. Each block can be processed independently in parallel.

## Block Manager

The Block Manager is responsible for managing and storing blocks in Spark. It handles the storage of RDDs, shuffle data, and other intermediate data in memory or on disk.

## Binning

Binning is a data preprocessing technique used to convert continuous variables into categorical variables by dividing the range of values into a series of intervals, or bins. This can help with data analysis and modeling.



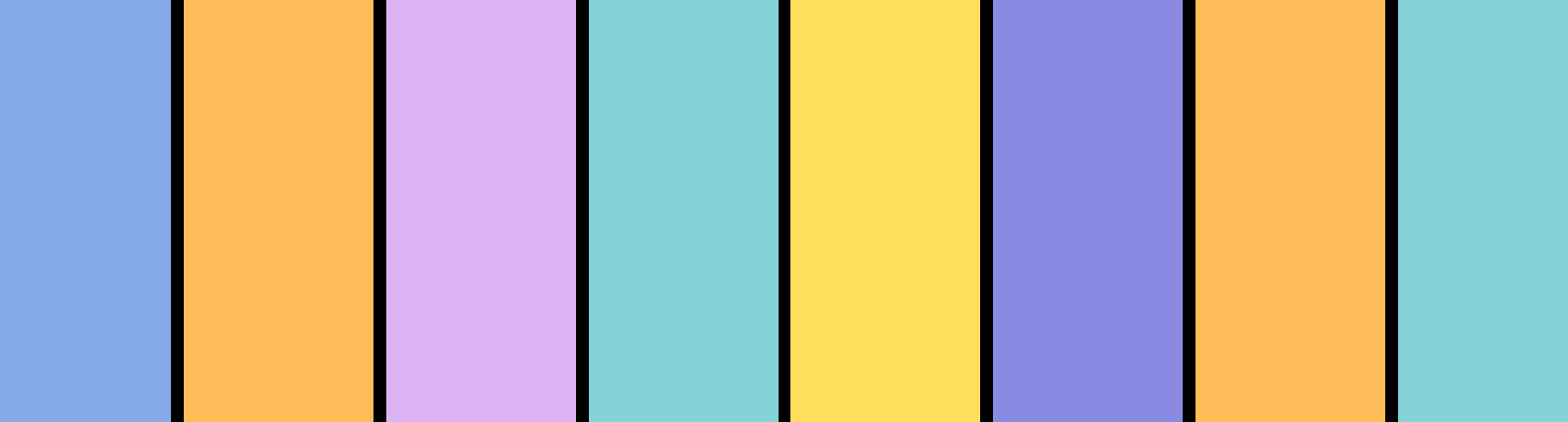
# B

## **BSP (Bulk Synchronous Parallel)**

BSP is a computational model that divides computations into supersteps, with each superstep consisting of local computation, communication, and barrier synchronization. Spark's execution model can be seen as loosely following the BSP model.

## **Back Pressure**

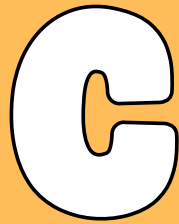
Back pressure is a mechanism used in Spark Streaming to handle situations where the data ingestion rate is higher than the data processing rate. It helps to control the flow of data to avoid overwhelming the system.



# B

## **Barrier Execution Mode**

Barrier execution mode in Spark allows for better coordination of stages that need to run concurrently. It is particularly useful for integrating with deep learning frameworks that require precise synchronization across all tasks



### **Caching**

The process of storing data in memory to speed up retrieval during processing.

### **Cluster Manager**

Manages resources in a cluster, such as YARN, Mesos, or Spark's standalone cluster manager.

### **Core**

The basic unit of parallelism in Spark. It is a thread running a task.





# C

## **Catalyst Optimizer**

The query optimization framework in Spark SQL that applies a series of optimization rules to improve the execution plan of queries.

## **Checkpointing**

A process of saving the intermediate state of an RDD or DataFrame to reliable storage, which is useful for long-running computations and fault tolerance.

## **Columnar Storage**

A data storage format that stores data in columns rather than rows, which is beneficial for analytical queries that typically access a subset of columns. Spark supports columnar formats like Parquet and ORC.

# C

## Coalesce

The `coalesce()` transformation is used to reduce the number of partitions in an RDD. It is more efficient than `repartition()` when reducing the number of partitions because it avoids a full shuffle.

## Column

In Spark SQL and DataFrame API, a column represents a single attribute or field in a dataset. Columns are used in expressions for transformations and aggregations.

## Compression

Compression refers to the process of reducing the size of data to save storage space and improve I/O performance. Spark supports various compression codecs, including Snappy, Gzip, and LZ4.

# D

<b>Dataframe</b>	A distributed collection of data organized into named columns, similar to a table in a relational database.
<b>Dataset</b>	A strongly-typed, immutable collection of objects that can be manipulated using functional transformations (map, flatMap, filter, etc.).
<b>Driver Program</b>	The main program that creates the SparkContext, connects to the cluster, and coordinates the execution of the tasks.

DEEPA

VASANTH KUMAR

# D

## **Delta Lake**

An open-source storage layer that brings ACID transactions to Apache Spark and big data workloads, enabling reliable data lakes with data versioning and time travel.

## **DAG (Directed Acyclic Graph)**

A representation of a sequence of computations to be performed on data. In Spark, a DAG is used to track the lineage of operations and optimize execution.

**DEEPA**

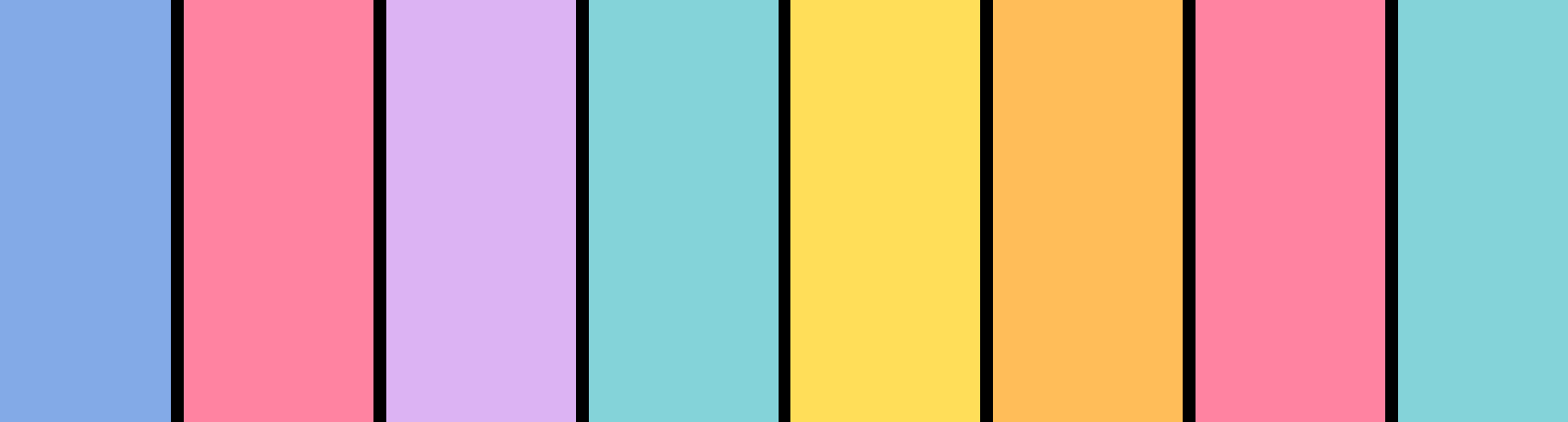
**VASANTH KUMAR**

# D

<b>Dependency</b>	Dependencies in Spark define the relationship between RDDs and are used to determine how data should be recomputed in case of a failure. There are two types of dependencies: narrow (e.g., map, filter) and wide (e.g., reduceByKey, join).
<b>Driver Node</b>	The driver node is the machine where the Spark driver program runs. It coordinates the execution of tasks on the worker nodes and maintains information about the Spark application's state.

DEEPA

VASANTH KUMAR



# D

## **Dynamic Allocation**

Dynamic allocation is a feature in Spark that allows the number of executors to be dynamically adjusted based on the workload. It helps in optimizing resource usage and cost.

## **Direct Stream**

In Spark Streaming, a direct stream is a type of DStream (Discretized Stream) that directly pulls data from a source such as Apache Kafka, ensuring exactly-once semantics and better fault tolerance.

## **Dynamic partition Pruning**

Dynamic partition pruning is an optimization technique in Spark that prevents scanning of unnecessary partitions when reading data.

# E

<b>Executor</b>	A distributed agent responsible for executing a task on a worker node and returning the results to the driver program.
<b>ETL (Extract, Transform, Load)</b>	A process in data warehousing and data integration that involves extracting data from source systems, transforming it to fit business needs, and loading it into a target data store.
<b>Event Time</b>	The time at which events actually occurred, as opposed to processing time when events are processed by the system. Spark Structured Streaming supports event-time processing.

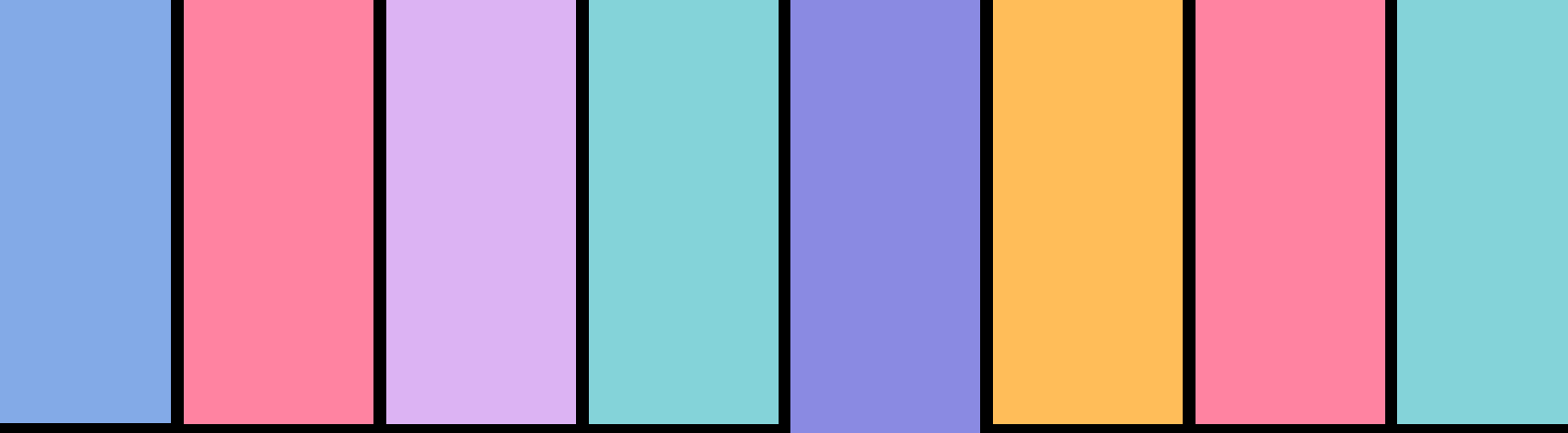
**DEEPA**

**VASANTH KUMAR**

# E

<b>Execution Plan</b>	The execution plan is a sequence of steps generated by the Spark Catalyst optimizer for executing a query. It includes physical and logical plans that describe how Spark will execute the query.
<b>Event Log</b>	Event logs are logs that Spark generates to record events such as job start, job end, task start, and task end. These logs can be used for monitoring and debugging Spark applications.
<b>Ephemeral Storage</b> <b>DEEPA</b>	Ephemeral storage refers to temporary storage that is used during the execution of a Spark application. It is not persistent and is typically used for intermediate data, such as shuffle files.





# E

## Edge Node

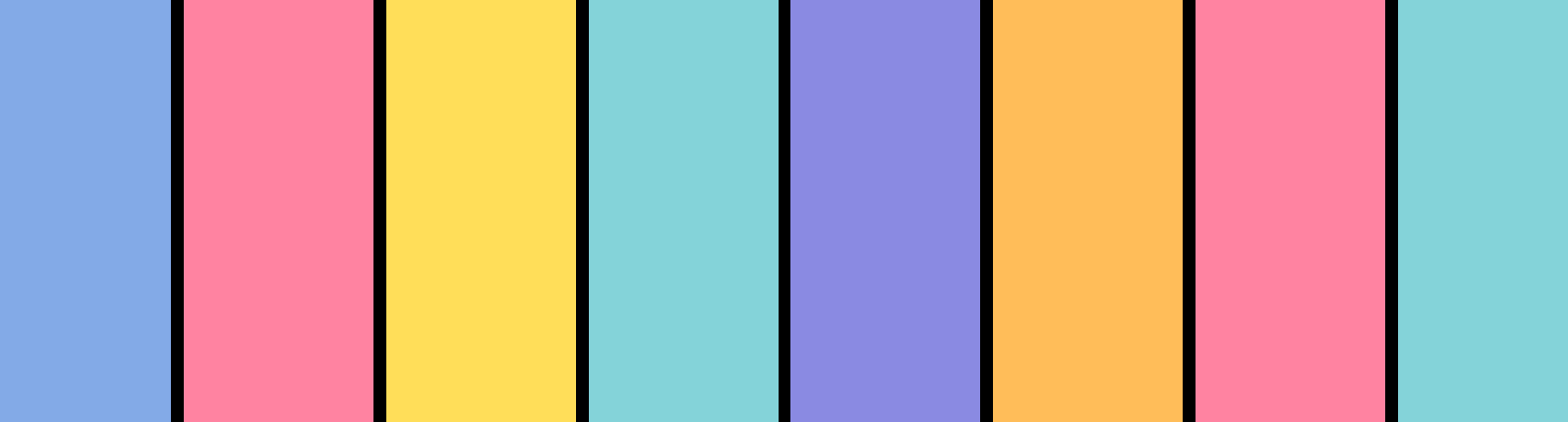
An edge node is a gateway node that sits between the user's local environment and the cluster. It is used to run client tools and host applications, such as Spark driver programs, that need to interact with the cluster.

## Execution Memory

Execution memory in Spark is the memory used for performing computations and storing intermediate results. It is distinct from storage memory, which is used for caching data.

## Execution Context

The execution context in Spark defines the environment in which Spark jobs run. It includes information about the cluster, resources, and configuration settings.



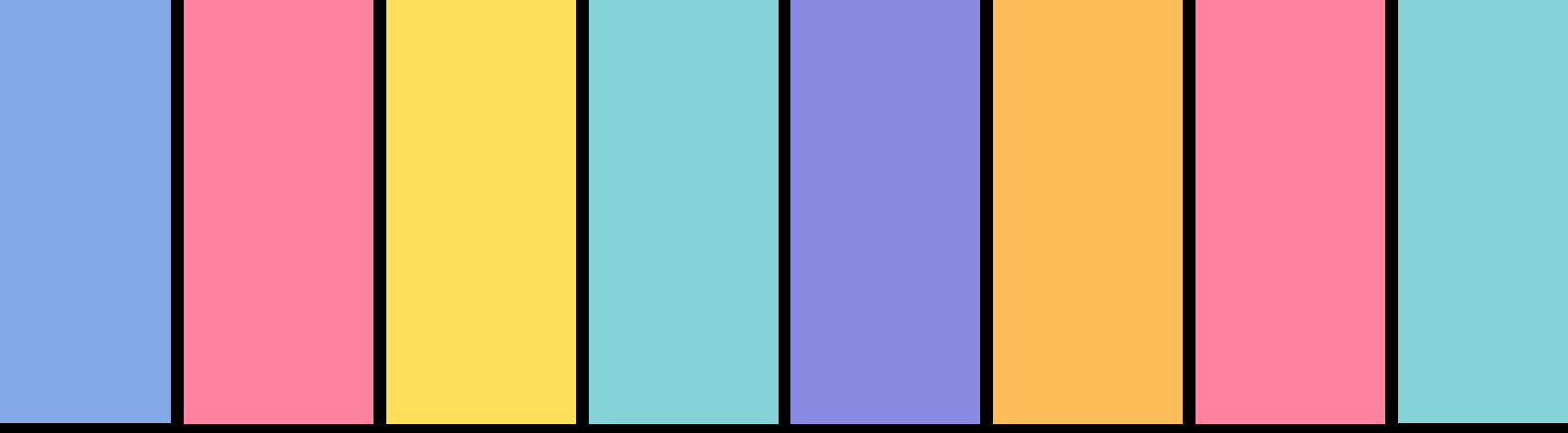
# F

<b>Fault Tolerance</b>	The ability of Spark to recover from node failures and recompute lost data.
<b>Functional Programming</b>	A programming paradigm in Spark where functions are treated as first-class citizens and operations are performed using transformations.
<b>FlatMap</b>	A transformation that applies a function to each element of an RDD or DataFrame and returns a new RDD or DataFrame by flattening the results.

FlatMap

DEEPA

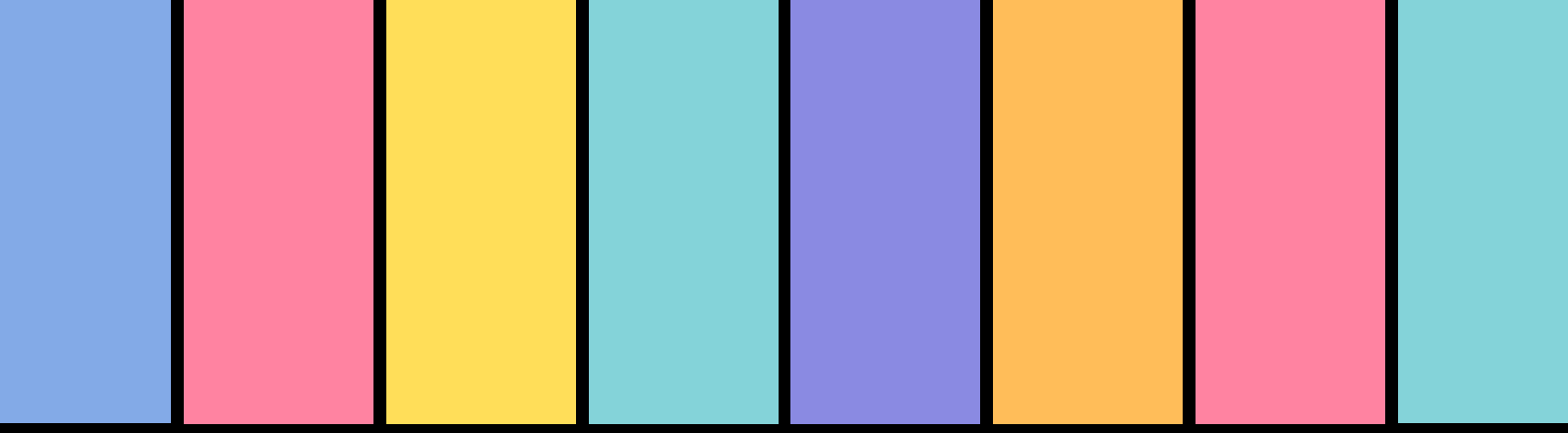
VASANTH KUMAR



# F

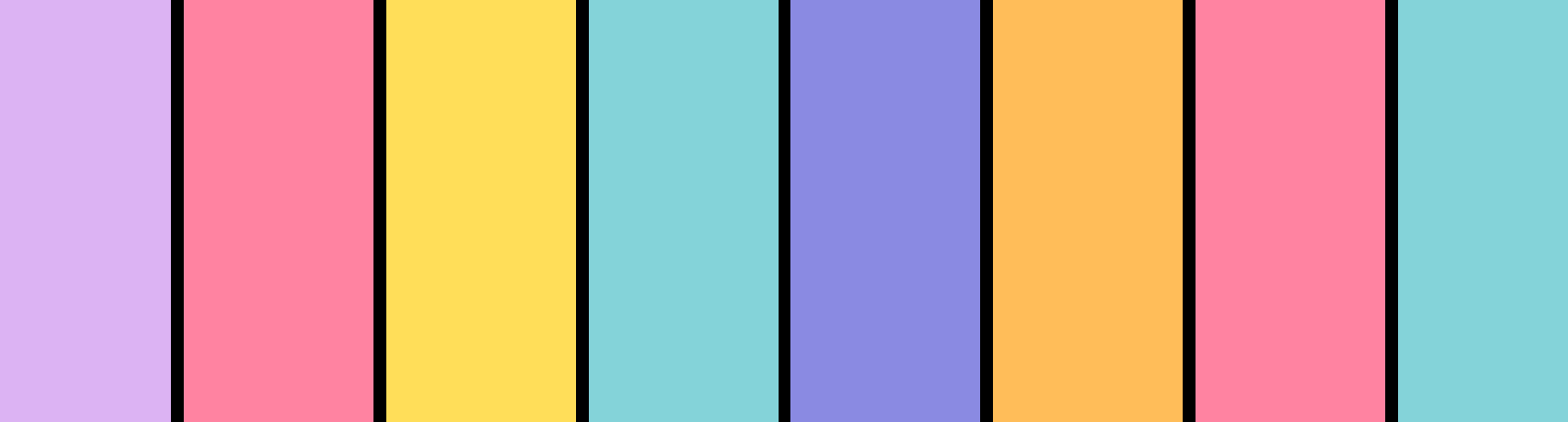
<b>Foreach</b>	The foreach action is used to apply a function to each element of the dataset. It is often used for side effects such as updating an accumulator or interacting with external storage.
<b>ForeachPartition</b>	Similar to foreach, foreachPartition applies a function to each partition of the dataset, allowing for more efficient processing of partition-level data.
<b>Fold</b>	The fold action is an aggregate function in Spark that combines elements of the dataset using an associative function and a neutral "zero value" to start the aggregation. It is similar to reduce but allows for a starting value.

FlatMap



# F

<b>Framework</b>	A collection of libraries and tools that provides a structured approach to building and managing applications. Spark is a big data processing framework.
<b>Filter</b>	A transformation that returns a new RDD or DataFrame containing only the elements that satisfy a given condition.



# G

## GraphX

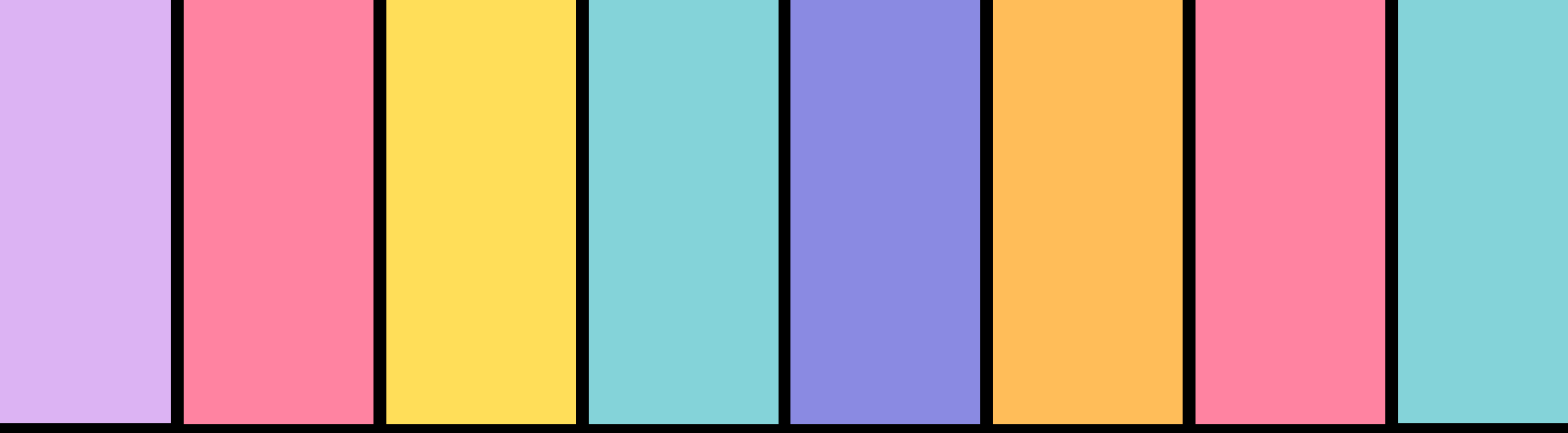
A component of Spark for graph processing and analysis.

## Gradient Descent

An optimization algorithm used in machine learning to minimize a function by iteratively moving towards the steepest descent.

## GroupByKey

A transformation that groups the values of a key-value pair RDD by key. It returns a new RDD of (key, Iterable<values>) pairs.



# G

<b>Global View</b>	A global view in Spark refers to viewing the entire dataset across all partitions, as opposed to looking at data within individual partitions.
<b>Garbage Collection</b>	Garbage collection in Spark refers to the automatic memory management process that reclaims memory occupied by objects that are no longer in use.

**DEEPA**

**SANTH KUMAR**

# H

<b>Hadoop</b>	An open-source framework for distributed storage and processing of large datasets. Spark can run on Hadoop clusters and use Hadoop's HDFS.
<b>Hive</b>	A data warehousing solution built on top of Hadoop that allows querying and managing large datasets using SQL. Spark can use Hive for reading and writing data.

DEEPA

VASANTH KUMAR

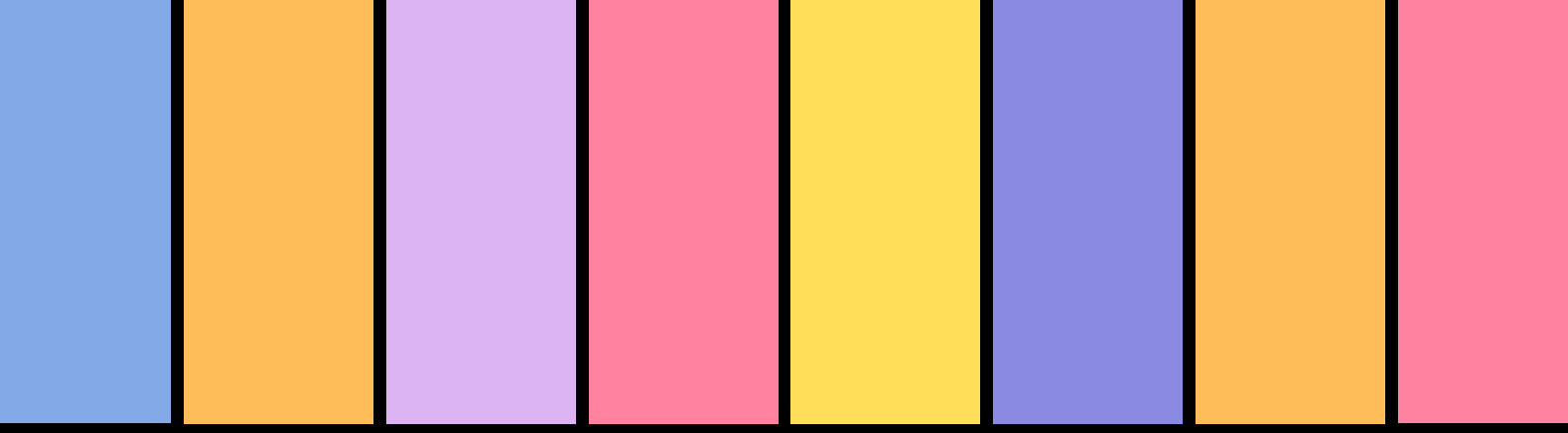
# H

<b>HDFS (Hadoop Distributed File System)</b>	A distributed file system designed to store large datasets across multiple nodes. Spark can read from and write to HDFS.
<b>HiveContext</b>	A class in Spark that allows querying data using the HiveQL language. It is part of the Spark SQL module and provides compatibility with Hive.

**DEEPA**

**VASANTHKUMAR**



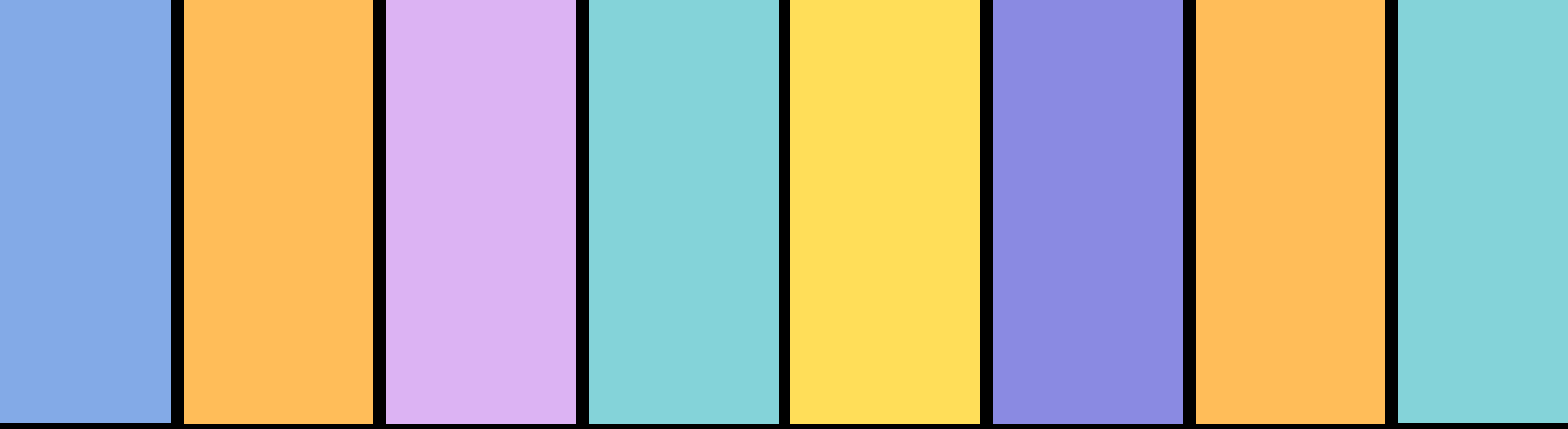


# H

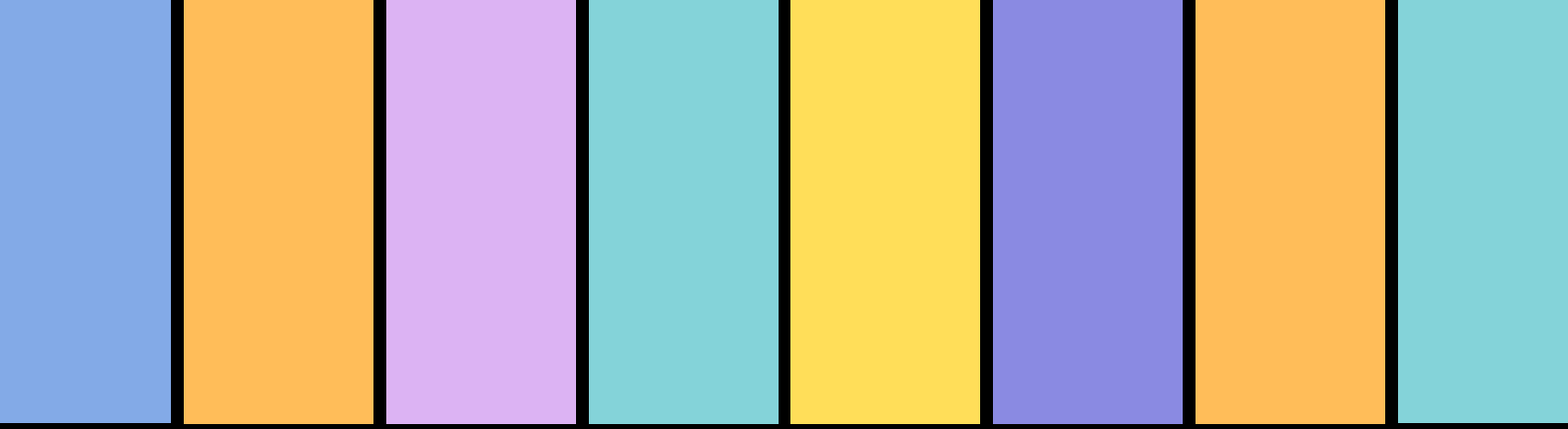
<b>Hash Partitioning</b>	Hash partitioning is a method of dividing data into partitions based on the hash value of keys. It is used in Spark to distribute data evenly across partitions for parallel processing.
<b>Hive Metastore</b>	A database that stores metadata about Hive tables, such as schema and location. Spark SQL can integrate with Hive Metastore to access this metadata.

DEEPA

VASANTHKUMAR



<b>In-memory Computing</b>	Storing data in memory to improve performance instead of reading from disk storage.
<b>Iterator</b>	An object in Spark that allows traversing through a collection of elements one at a time.
<b>InputFormat</b>	A class in Hadoop (and used by Spark) that defines how input data is split and read into the system. Examples include TextInputFormat and SequenceFileInputFormat.



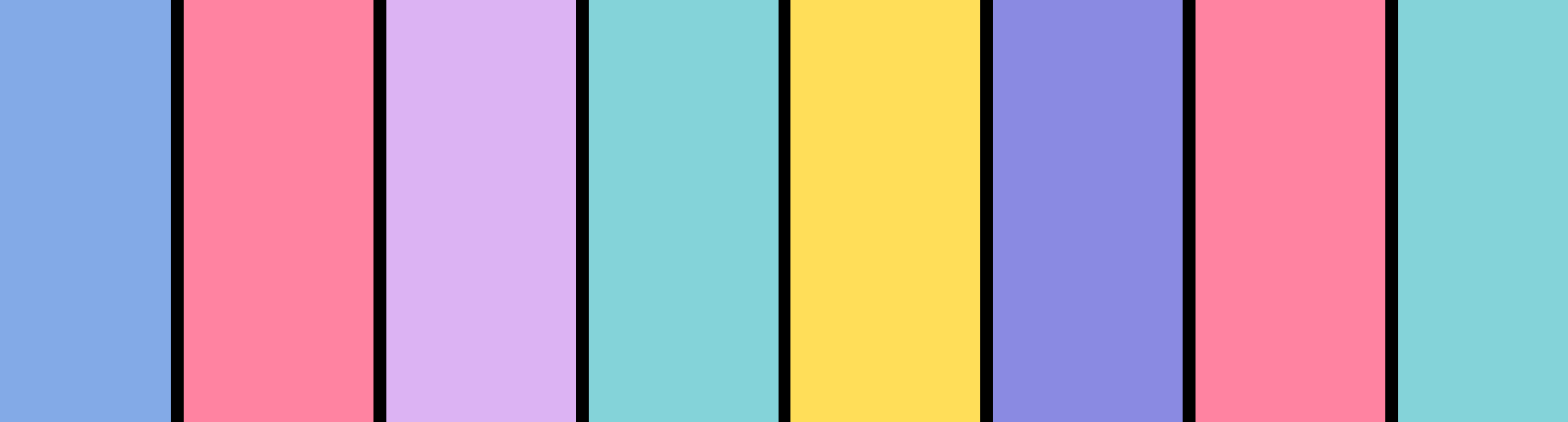
<b>Interactive Query</b>	Interactive query in Spark allows users to perform ad-hoc queries on large datasets and get results quickly. Spark SQL and DataFrames are often used for interactive querying.
<b>Immutable</b>	In Spark, RDDs (Resilient Distributed Datasets) are immutable, meaning once created, they cannot be changed. This immutability provides consistency and simplifies parallel processing.

# J

<b>Job</b>	A sequence of tasks submitted to the cluster for execution, generated by a Spark action.
<b>Join</b>	A transformation that combines two RDDs or DataFrames based on a common key. Types of joins include inner join, outer join, left join, and right join.
<b>Job Scheduler</b>	The component of Spark that handles the scheduling of jobs, dividing them into stages and tasks, and distributing them across the cluster for execution.

DEEPA

VASANTH KUMAR



# J

<b>Java Virtual Machine (JVM)</b>	JVM is the virtual machine that runs Java bytecode and other JVM-based languages. Spark runs on the JVM and uses it for executing its tasks and applications.
<b>JobServer</b>	JobServer in Spark is a service that manages and runs Spark jobs. It provides an interface for submitting jobs, monitoring their execution, and managing job resources.

DEEPA

VASANTH KUMAR

# K

## Kryo

A serialization library used by Spark for fast and efficient serialization of objects.

## Kafka

A distributed streaming platform that Spark can integrate with for processing real-time data streams.

## Kinesis

A real-time data streaming service provided by AWS. Spark can read data from and write data to Kinesis streams for real-time data processing.

**DEEPA**

**VASANTH KUMAR**

# K

<b>Key</b>	In Spark, a key typically refers to the attribute or field used to partition or organize data in key-value pair RDDs or DataFrames. Operations like <code>groupByKey</code> and <code>join</code> are based on keys.
<b>Kurtosis</b>	Kurtosis is a statistical measure of the "tailedness" of the probability distribution of a dataset. Spark MLlib provides functions for calculating kurtosis as part of statistical analysis.

**DEEPA**

**VASANTH KUMAR**



<b>Lazy Evaluation</b>	A technique used in Spark where transformations on RDDs are not immediately executed but are recorded in a lineage graph for optimization.
<b>Lineage</b>	A record of the transformations applied to an RDD, used for fault tolerance and recomputation.
<b>Logical Plan</b>	An abstract, high-level representation of a query that describes what operations need to be performed. Spark's Catalyst Optimizer generates and optimizes logical plans before converting them into physical plans for execution.

**DEEPA**





<b>Livy</b>	Livy is a REST service for Apache Spark that allows remote applications to submit Spark jobs, monitor their status, and retrieve results programmatically. It simplifies integration with Spark clusters.
<b>Local Mode</b>	Local mode in Spark refers to running Spark applications on a single machine using a single JVM process, without utilizing a distributed cluster. It is useful for testing and development.
<b>Levenshtein Distance</b>	Levenshtein distance is a metric used in Spark SQL and String manipulation for measuring the difference between two sequences. It is useful for fuzzy matching and similarity checks.

# M

## MapReduce

A programming model for processing large datasets. Spark can execute MapReduce tasks much faster due to its in-memory computing capabilities.

## MLlib

A machine learning library in Spark providing various algorithms and utilities for scalable machine learning.

DEEPA

VASANTHKUMAR

# M

<b>Map</b>	A narrow transformation that applies a function to each element of an RDD or DataFrame, returning a new RDD or DataFrame with the results.
<b>MapPartitions</b>	A transformation that applies a function to each partition of an RDD or DataFrame, rather than to each element. This can be more efficient for certain operations.

**DEEPA**

**VASANTHKUMAR**

# M

<b>Mesos</b>	Apache Mesos is a cluster manager that can dynamically share resources across multiple Spark applications. Spark can run on Mesos, leveraging its resource isolation and sharing capabilities.
<b>Memory Management</b>	Spark manages memory usage through different storage levels (e.g., MEMORY_ONLY, MEMORY_AND_DISK) and memory management policies to optimize performance and resource utilization.

DEEPA

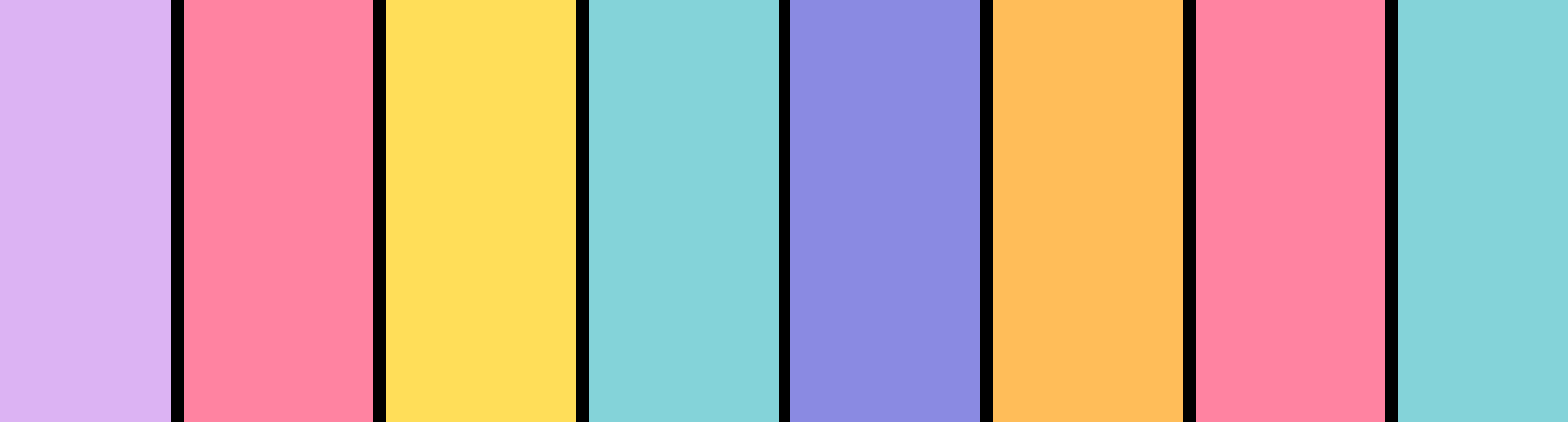
VASANTH KUMAR

# M

<b>Master Node</b>	The master node in Spark is the node that hosts the SparkContext, coordinates the execution of tasks across worker nodes, and manages the overall execution of Spark jobs.
<b>Micro-batching</b>	Micro-batching is a streaming processing technique used in Spark Streaming where data is processed in small, finite-sized batches. It balances low-latency processing with efficient resource utilization.

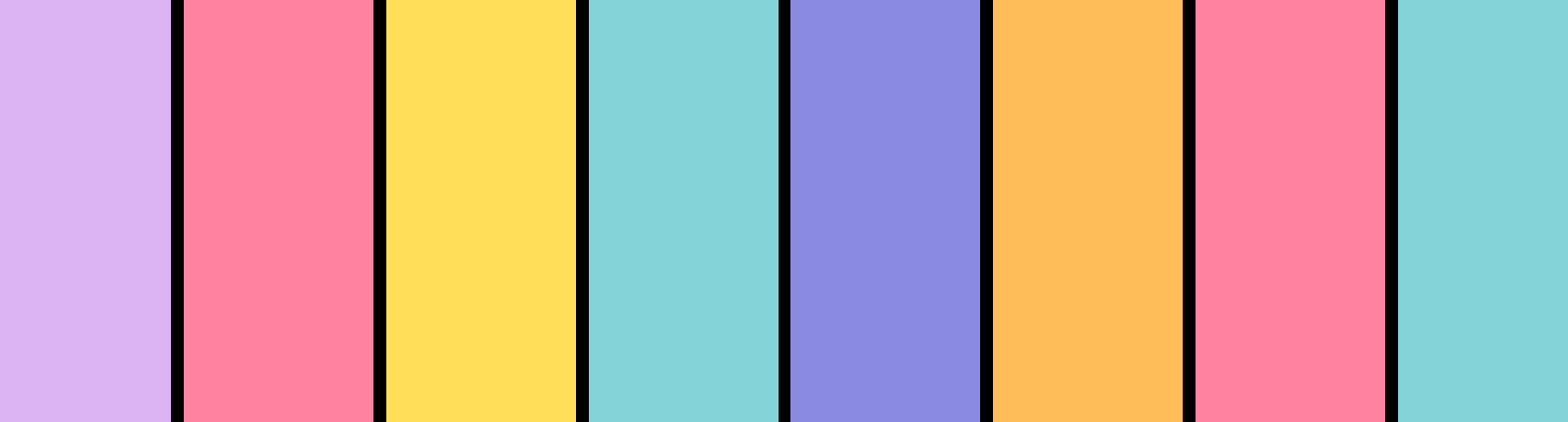
DEEPA

VASANTHKUMAR



# N

<b>Node</b>	A single machine in a Spark cluster that can be a driver or a worker.
<b>Notebook</b>	An interactive environment, such as Databricks Notebooks or Jupyter Notebooks, where users can write and execute Spark code, visualize data, and document their analysis.
<b>Normalization</b>	The process of structuring data to reduce redundancy and improve data integrity. In Spark, this often involves transforming and cleaning data before analysis.



# N

## **Narrow Dependency**

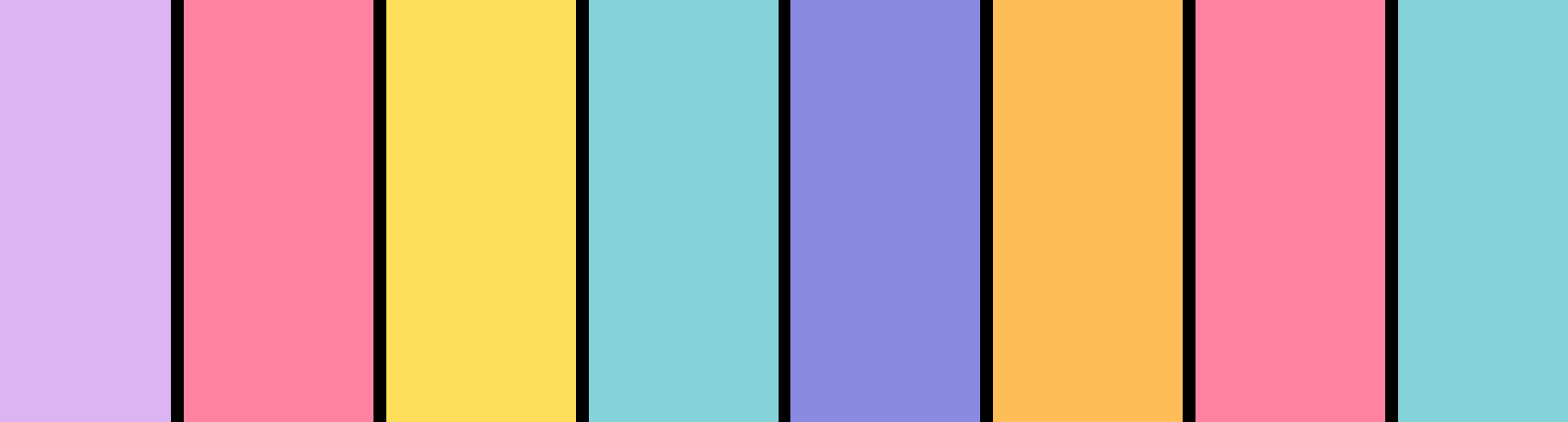
Narrow dependency (or narrow transformation) in Spark refers to transformations like map, filter, or flatMap that do not require data to be shuffled across partitions, making them more efficient.

## **NaN**

NaN (Not a Number) is a special floating-point value used to represent undefined or unrepresentable numerical results. Spark handles NaN values in computations and transformations.

## **Namespace**

In Spark SQL and Hive, a namespace is a logical container for tables, views, and other database objects. Namespaces help organize and manage data assets within a data catalog.



# N

## NodeManager

NodeManager is a component of Apache YARN (Yet Another Resource Negotiator) that runs on worker nodes in a Hadoop cluster. It manages resources (CPU, memory) and executes tasks allocated by the ResourceManager, including Spark tasks.

## Numpy

NumPy is a numerical computing library for Python. Spark integrates with NumPy to leverage its capabilities for efficient array operations and numerical computations within Spark applications.



0

<b>Operations</b>	Functions in Spark that can be applied to RDDs, such as transformations and actions.
<b>Optimization</b>	Techniques used to improve the performance of Spark jobs, including query optimization and execution plan tuning.
<b>OutputFormat</b>	A class in Hadoop (and used by Spark) that defines how the output data is written. Examples include TextOutputFormat and SequenceFileOutputFormat.

DEEPA

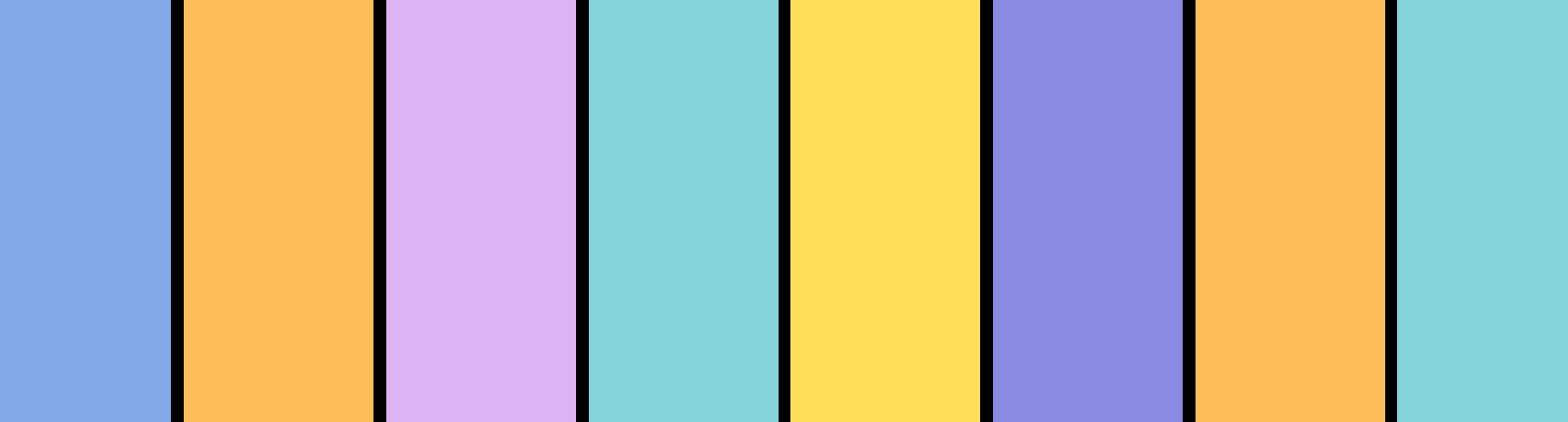
VASANTH KUMAR

0

<b>Overhead</b>	Overhead in Spark refers to the additional resources (such as CPU and memory) consumed by Spark framework operations, beyond what is directly used by user tasks.
<b>Out-of-Core Processing</b>	Out-of-core processing in Spark refers to the ability to process data that is larger than the available memory by spilling data to disk, enabling efficient handling of big data.

DEEPA

VASANTH KUMAR



# P

## **Partition**

A division of data in an RDD or DataFrame that can be processed in parallel.

## **Persist**

Storing an RDD in memory across operations for faster access.

## **PySpark**

The Python API for Spark, allowing the use of Spark with Python.



# P

<b>Pipeline</b>	A sequence of data processing steps, often used in machine learning workflows. Spark's MLlib provides APIs to create and manage pipelines.
<b>Physical Plan</b>	A detailed, low-level representation of how a query will be executed in Spark. It is generated by the Catalyst Optimizer from the logical plan.



# P

## **Predicate**

A predicate in Spark is a function used to filter data based on a specified condition, such as in filter operations on RDDs or DataFrames.

## **PairRDDFunction s**

PairRDDFunctions are specialized functions in Spark's Scala API for operations on RDDs containing key-value pairs, providing methods like `reduceByKey`, `groupByKey`, and `join`.

## **Parquet**

Parquet is a columnar storage format supported by Spark for efficient data storage and query processing. It offers benefits such as compression and schema evolution support.

# Q

<b>Query</b>	A request for data retrieval and processing in Spark, often written in SQL or using DataFrame/Dataset APIs.
<b>Query Execution Plan</b>	The series of steps Spark takes to execute a query, which includes both the logical plan and the physical plan.

**DEEPA**

**VASANTHKUMAR**



# R

<b>RDD (Resilient Distributed Dataset)</b>	The fundamental data structure in Spark, representing an immutable, distributed collection of objects.
<b>Resource Manager</b>	Manages the allocation of resources in a cluster for Spark jobs.
<b>Range Partitioning</b>	A partitioning strategy where data is divided into ranges based on a key. This can optimize performance for range queries and joins.

**DEEPA**

**VASANTH KUMAR**

# R

<b>ReduceByKey</b>	A wide transformation that merges the values of each key using an associative reduce function. It is more efficient than groupByKey because it performs partial aggregation locally before shuffling the data.
<b>Repartition</b>	A transformation that reshuffles the data in an RDD or DataFrame into a specified number of partitions. This can be used to increase or decrease the level of parallelism.
<b>ROW</b>	A record in a DataFrame, representing a single entry with fields corresponding to the columns of the DataFrame.

DEEPA

VASANTH KUMAR

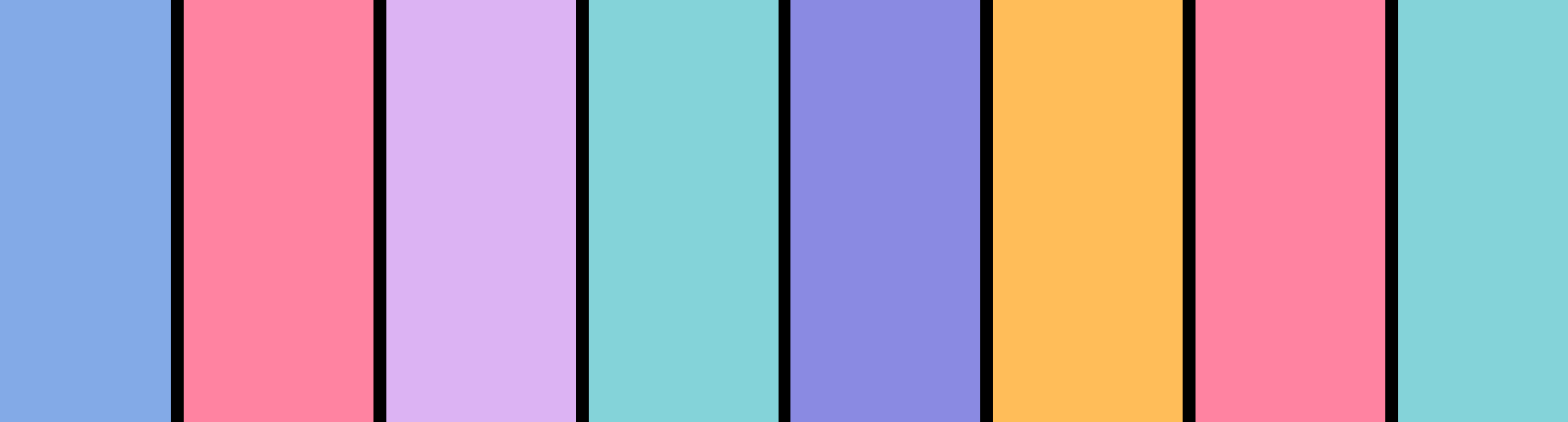


# R

<b>Replication</b>	Replication in Spark refers to the process of duplicating data or RDD partitions across nodes in a cluster to achieve fault tolerance and data locality.
<b>Resilience</b>	Resilience in Spark refers to its ability to recover from failures or faults automatically by using lineage information and recomputing lost data partitions.
<b>Resource Manager</b>	Resource Manager, such as YARN (Yet Another Resource Negotiator), manages and allocates resources (CPU, memory) for Spark applications running in a cluster environment.

DEEPA

VASANTH KUMAR

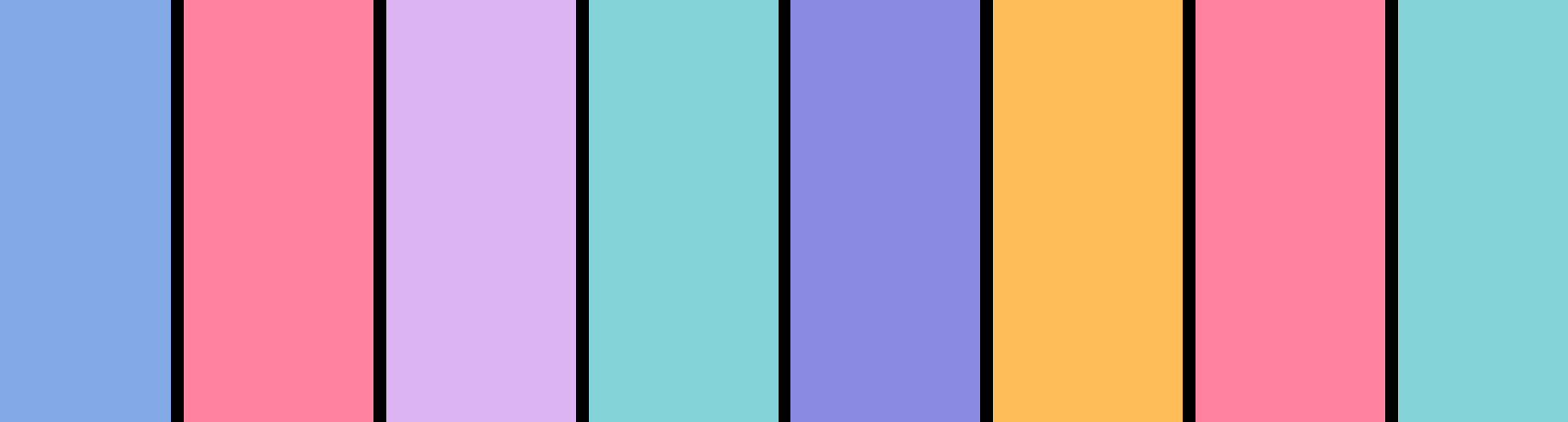


# S

<b>SparkContext</b>	The main entry point for Spark functionality, responsible for connecting to the cluster and creating RDDs.
<b>Spark SQL</b>	A Spark module for working with structured data using SQL queries.
<b>Streaming</b>	Processing real-time data streams in Spark, using the Spark Streaming API.

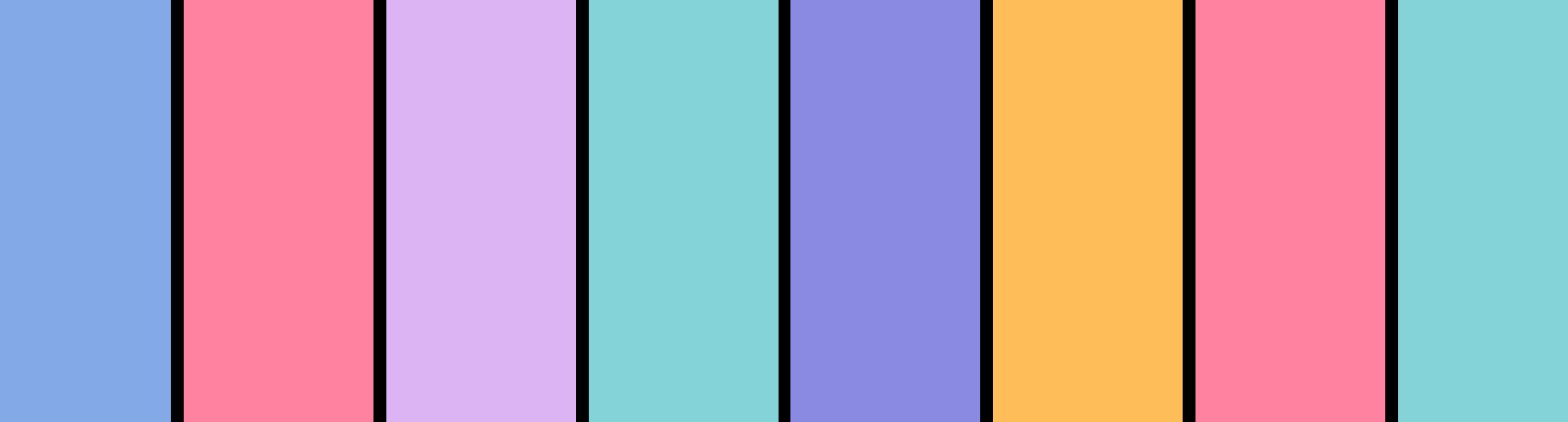
**DEEPA**

**VASANTH KUMAR**



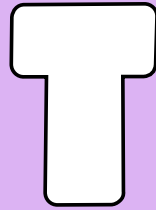
# S

<b>Schema</b>	The structure that defines the organization of data in a DataFrame or Dataset, including column names and data types.
<b>Shuffle</b>	A process of redistributing data across partitions that involves moving data between executors. It typically occurs during wide transformations like <code>groupByKey</code> , <code>reduceByKey</code> , and <code>join</code> .
<b>SparkSession</b>	The entry point for programming Spark applications with the DataFrame and Dataset API. It replaces the older <code>SQLContext</code> and <code>HiveContext</code> .



# S

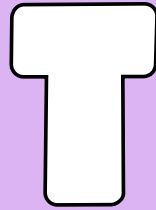
<b>Stage</b>	A set of tasks that can be executed in parallel during the execution of a Spark job. A job is divided into stages based on shuffle boundaries.
<b>Structured Streaming</b>	An API in Spark for stream processing that allows you to process data in real-time using high-level declarative queries similar to batch processing.
<b>Salting</b>	Salting in Apache Spark is a technique used to address data skew issues when performing certain operations, particularly joins and aggregations, on large datasets. Data skew occurs when the distribution of data across partitions is highly uneven, causing some partitions to contain significantly more data than others.



<b>Task</b>	A unit of work that runs on a single executor and is a part of a job.
<b>Transformation</b>	Operations that create a new RDD from an existing one, such as map, filter, and reduceByKey.
<b>Triggers</b>	A mechanism in Structured Streaming that specifies when the system should process the next set of data. Examples include continuous processing and micro-batch intervals

**DEEPA**

**VASANTH KUMAR**



<b>Task Scheduler</b>	The task scheduler in Spark coordinates the assignment of tasks to executor nodes in a cluster, optimizing task placement based on data locality and resource availability.
<b>Tuple</b>	In Spark, a tuple is an ordered collection of elements (similar to a list or array) that can contain heterogeneous data types, commonly used to represent key-value pairs in RDDs or DataFrames.
<b>Tungsten</b>	Tungsten is the internal code name for a Spark project focused on optimizing memory management and execution speed by leveraging binary processing and off-heap memory.

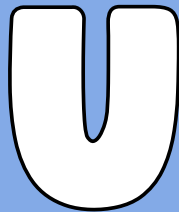
# U

<b>UDF (User-Defined Function)</b>	Custom functions defined by users to extend the capabilities of Spark SQL.
<b>Unpersist</b>	Releasing the memory used by a cached RDD.
<b>Unit Tests</b>	Tests that validate the functionality of individual components of Spark applications. Libraries like spark-testing-base can help write unit tests for Spark applications.

DEEPA

Unit Tests

VASANTH KUMAR



<b>Union</b>	union is a transformation in Spark that combines two RDDs or DataFrames into a single RDD or DataFrame by appending the rows or columns.
<b>Upstream Dependency</b>	Upstream dependency in Spark refers to the dependencies of a task or operation on previous stages or transformations in the execution DAG (Directed Acyclic Graph).





V

<b>View</b>	A temporary table in Spark SQL created from a DataFrame.
<b>Vectorization</b>	A technique used in Spark MLlib to represent data in a format that can be efficiently processed by machine learning algorithms.

DEEPA

VASANTH KUMAR



# W

## **Worker Node**

A node in a Spark cluster that executes tasks and returns results to the driver.

## **Wide Transformation**

Transformations that require data shuffling across nodes, such as `groupByKey` and `reduceByKey`.

## **Window Function**

A function that performs a calculation across a set of table rows related to the current row. Spark SQL supports window functions for operations like ranking, aggregations, and analytic functions.

# W

## **Write-ahead Log (WAL)**

WAL in Spark Streaming is a mechanism for fault tolerance, storing data about processed records in a log before they are committed to a data source, ensuring recovery in case of failures.

## **Widening Conversion**

In Spark's type system, widening conversion refers to automatic type promotion or casting of data types to a wider or more general type to maintain compatibility and avoid data loss.



# X

<b>XML</b>	A markup language that Spark can read and write using libraries and custom parsers.
<b>XPath</b>	XPath is a query language used for selecting nodes from XML documents, which can be integrated with Spark applications for data extraction and processing.

**DEEPA**

**VASANTHKUMAR**

# Y

<b>YARN (Yet Another Resource Negotiator)</b>	A cluster management technology used by Spark for resource allocation and job scheduling.
<b>YAML:(Yet Another Markup Language)</b>	YAML ain't markup language (a recursive acronym), which emphasizes that YAML is for data, not documents is a human-readable data serialization format, which can be used for configuring and managing settings in Spark applications or related tools.

DEEPA

VASANTHKUMAR



# Z

## **Zookeeper**

A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services, often used in Spark streaming for managing offsets in Kafka.

## **Z-Order**

A multidimensional clustering method used in Delta Lake to optimize data skipping and improve query performance by clustering data in a way that enhances locality for multiple columns.

## **Zeppelin**

An open-source web-based notebook that enables interactive data analytics. Apache Zeppelin supports multiple languages and can be used with Spark for interactive data exploration and visualization.



**DEEPA VASANTHKUMAR**



<https://www.linkedin.com/in/deepa-vasanthkumar/>



<https://medium.com/@deepa.account>