Scala → object oriented + function Programming feautures.

Spark SQL ↓ handles structured or semi-structured

Day / Date

Supports data analysis/ML/streaming data/graphs.
↑

APACHE SPARK (fast & general engine for large-scale data)

↓

programming operations → RDD.

- Written in Scala
- Developed by AMPlab UC Berkely, now by Databricks.com

Spark Core API				
R	Python	Scala	SQL	Java.

- Spark Arthitecture
  - Master-Slave arthitecture → master → 'Driver'
    slaves → 'Workers'
  - Transformation and actions are executed on worker node.
- Every spark application requires a spark context (main entry point to spark AI)
- Spark shell provides preconfigured spark context called sc.

- RDD (Resilient Distributed Dataset)

  Resilient ↓ data in memory lost, it can be recreated

  Distributed ↑ processed across cluster

  - RDD fundamental units of data in spark (date contained)
  - In-memory computation → Computed results stored in distributed memory (RAM). Very fast.
  - Lazy Evaluation → · not performed immediately
    · action than
  - Fault tolerance → · If failure occurs inany partition of RDD, partition can be re-computed from orignal fault tolerance input data to create it.

- A full Spark API can be used with Spark SQL data by accessing underlying RDD.

Day / Date Calling persist or cache does not trigger execution / computation.

- o— Immutability → value can not be Changed
- o— Partitioning → • collection of various data (RDD)
  - Can not fit into a single node
- o— Persistance → • Save result of RDD evaluation
  - Stores intermediate result.

Transformation	Action
• define new RDD based on Current one	return values. to driver / master node.
• map ( ) → returns collection	• count ( ) takesample( )
filter ( )	returns first n rows ← take (n) take ordered( )
flatmap ( ) → returns flattened result.	collect all the rows. ↙ collect ( )
	Save as Textfile ( )
	reduce ( ) Saveasobjectfile( )
	first ( ) Countbykey ( )
	Saveas SequenceFile ( )
	foreach ( )

input item can be mapped to 0 or more output (returns Seq rather than single) ← flatmap ( )

- RDD organized as Directed Acyclic graph (DAG)

DAG track dependencies (Linease)

- •-nodes are RDDs
- •-arrows are transformation

2.

- Pipelining (Spark will perform sequences of transformation by row so no data is lost).

↓

action reexecutes the lineage transformation starting with base.

- RDD Lineage (Spark maintains each RDD lineage- previous RDD on which it depends)

- Spark performance Tuning (• computations over RDD, transformation embedded in chain)
  - dataset loaded from databases Computation are performed results are returned.

Q:- Call actions on RDD?

- recomputation of transformations each time, increase resource usage. To optimize, see that action is performed again and again.

JVM → Java virtual machine.

Day / Date

Cache method is implemented itself as a call to rdd.persist (StorageLevel. •Memory_only)
↑

RDD: cache()

- Cache RDD into memory
- By calling cache method, first action say RDD keep values it has calculated in memory. RDD than uses cached values for calculating 2nd action.

Unpersisting

- As more and more RDD's are cached, memory decrease.
  - Spark starts expelling partitions from cache.
  - Zaida JVM garbadge collection time → Unavoidable.
- Call un-persist method on RDD when caching is no longer needed.

Why not to use Caching (Recomputation is faster as increase memory → more money.)

↓ if once to read dataset then no point of caching

↓ depends on
- How many times data is accessed
- Amount of work involved.

PySpark (interface for spark in python)

↓ pyspark shell.

Spark Dataframe	Pandas Dataframe.
• Supports parallelization	no parallelization
• Multiple nodes	single node
• Lazy execution	Eager execution
• Immutable	Mutable
• Distributed & Spark dataframe is faster for large amount of data.	not distributed & slow for large data.

Maxim

- Simple Algo for frequent elements in stream & bags

Day / Date

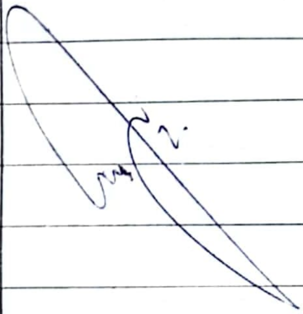
- Spark SQL is not a replacement for a database ~~ETL/structured to other application.~~

Narrow Transformation	Wide Transformation
each input partition → one output partition	each input → many outputs
Faster	Slower
Not require any data shuffling over cluster network	might require data shuffling over cluster network.
Map, flatten map, Mappartition, filter, sample, Union	Intersection, Join, Cartesian, repartition

reduce by key
- returns dataset of (K, V) pairs where values for each key are aggregated.
- map side combine
- same to combiner in map reduce().

groupby key (returns dataset of (K, iterables) pairs.
- do not map side combine)

↓ cause diverse effect to output.



- **Spark Dataframes.** (main abstraction in Spark SQL)
  - Comparable to RDDs in Core Spark.
  - Distributed collection of data organized into named columns.
  - are Created →
    - existing structured data source
    - existing RDD
    - programmatically defining a schema.

Example: entry point

```
from pyspark.sql import SQLContext
# Intilize sparksession
spark = SparkSession.builder \
    .appname ("Tashi") \
    .getOrCreate( )
or
sqlC= SQLContext (sc)
```

sqlCtx.sql("Select ....)

Day / Date

deals with dataframe metadata

• **Dataframe Basic Operations.**

- schema – schema object describing data.
- printSchema – diplays schema as visual tree
- Cache/persist
- Columns (names of columns)
- dtypes – array of (col name, type) pairs.
- explain – prints debug information about dataframe.

• **DataFrame Queries.** (returns new dateframe similar to RDD transformation)

- distinct – returns new dataframe with different elements.
- join – 1 dataframe join with another
- limit –
- select –
- filter –

```
DF.select ("age")
DF.where (" age>21")
```

- Some queries take one or more cols

```
ageDF = DF.select (DF.age)
DF.select (DF.name, DF.age+10)
DF.sort (DF.age.desc())
```

- frequent Pattern mining => df.freqItems.
- Data can be stored to date source.
  - Built in support for JDBC & Parquet file
  - Create JDBCTable
  - Insertinto
  - SaveAsParquetFile
  - SaveAsTable
- Dataframes are built on RDDs.
  - Base RDDs contains row object.
  - use rdd to get underlying rdd.
- Row RDDs have all standard actions and transformations.