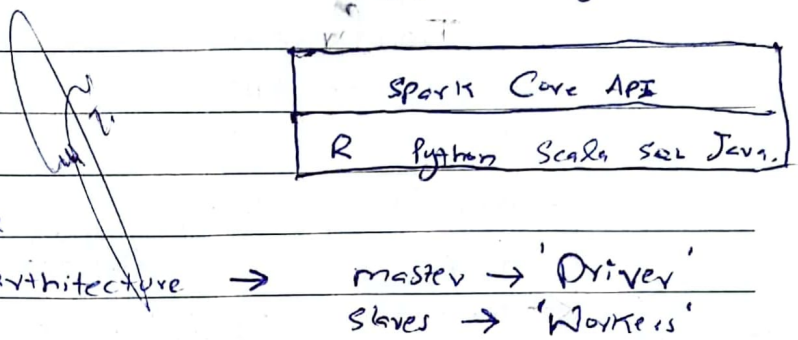
Tashfeen Abbasi/Laiba Mazhar
122-2041/122-1855

Day/Date

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 Architecture
  - Master-Slaves architecture → master → 'Driver'
    slaves → 'Workers'
  - Transformation and actions are executed on worker node.
- Every spark application requires a spark context (main entry point to spark API)
- 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
  - Dataset: data container
  - RDD fundamental units of data in spark
  - In-memory computation → Computed results stored in distributed memory (RAM). Very fast.
  - Lazy Evaluation → not performed immediately action than
  - Fault tolerance → If failure occurs in any partition of RDD, partition can be re-computed from original fault tolerance input data to create it.

Maxim.....

Spark SQL → handles structured or semi-structured

Supports data analysis / ML / streaming / graphs. data

Function Programming features.

object oriented +

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

Day / Date

Calling persist or cache does not trigger execution / computation.

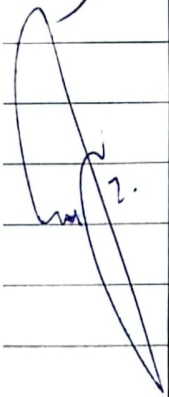
- Immutability → value can not be changed
- Partitioning → • collection of various data (RDD)
  • can not fit into a single node
- Persistance → • save result of RDD evaluation
  • Stores intermediate result.

Transformation
- define new RDD based on current one
- map( ) → returns collection
- filter( )
- flatmap( ) → returns flattened result.

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

Action
Return values to driver / master node.
- count( ) ← returns first n rows ← take(n)
- collect( ) ← collect all the rows.
- saveAsTextFile( )
- reduce( )
- first( )
- saveAsSequenceFile( )
- foreach( )
- takeSample( )
- takeOrdered( )
- saveAsObjectFile( )
- countByKey( )

- **RDD organized as Directed Acyclic Graph (DAG)**
  DAG track dependencies (Lineage)
  - nodes are RDDs
  - arrows are transformation



- **Pipelining** (Spark will perform sequences of transformation is just by row so no data)
  ↓
  action re-executes 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 database
  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.

Maxim

JVM → Java virtual machine.

Day / Date

Cache method is implemented itself as a call the 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 partition from cache.
  - Zeida JVM garbage 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 streams & bags

Day / Date

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

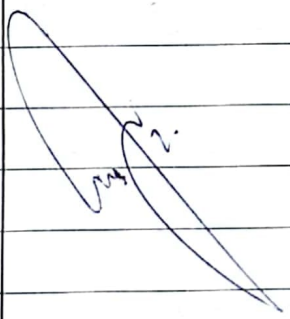
- **Narrow Transformation** — **Wide Transformation**

- each input partition → one output partition
- each input → many outputs

- Faster
- Not require any data shuffling over cluster network

Slower might require data shuffling over cluster network.

- Map, flatten map, Map partition, filter, sample, union
- Intersection, Join, Cartesian, repartition



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

reduce by key / **groupby key** returns dataset of (K, iterable<V>) 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
    - programatically defining a schema.

Example:

entry point

```
from pyspark.sql import SQLContext
# Initialize SparkSession
spark = SparkSession.builder
    .appName("Tashi")
    .getOrCreated()
or
sqlC = SQLContext(sc)
```

Maxim..

sqlCtx.sql("Select ....")

Day / Date

deal with dataframe metedata

- **Dataframe Basic Operations.**
  - schema – schema object describing data
  - printSchema – displays schema as visual tree
  - cache/persist
  - columns (names of columns)
  - dtypes – array of (col name, type) pairs.
  - explain – prints debug infoymation about dataframe.
- **DataFrame Queries.** (returns new dataframe. Similar to RDD transformation)
  - distinct – return 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

1.
```
ageDF = DF.select(DF.age)
DF.select(DF.name, DF.age+1)
DF.sort(DF.age.desc())
```

- Frequent Pattern mining => df.freqItems.
- Data can be stored to data source.
  - Built in support for JDBC & Parquet file
  - Create JDBCTable
  - Insrytinto
  - SaveasParquetFile
  - Save As table
- Dataframes are built on RDDs.
  - Base RDDs contains row object.
  - use rdd to get underlying rdd.
- Row RDDs have all standard action and transformations

Maxim.....