Recap:

Streaming :

Spark 👍

Multiple libraries /processing : ML,streaming,batch,graph

Polyglot : scala /java/python/R/SQL

Cluster : Spark standalone , YARN , Mesos , K8s

Data source / target : diff source / target

======================================

RDD -Resilient Distributed Datasets

RDD → collection of partitions

Characteristic :

1. Immutable
2. Type inference
3. Cacheable
4. Fault tolerance
5. Lazy evaluation

Operations :

Transformation : new rdd → flatmap,map,filter

Actions : Triggers the job

Collect, count, saveAsTextFile

=======================================

Transformation : Narrow and Wide

Emp.csv b1 b2 b3 => HDFS

1-1000 1001-2000 2001-3000

Emprdd p1 p2 p3 ⇒ source rdd

1-1000 1001-2000 2001-3000

Emp\_fil\_mgr\_rdd => filter rdd

P1 p2 p3

20 20 20

Emp\_top3\_mgr\_rdd ⇒rank rdd

P1

3

emp\_top3\_mgr\_rdd.collect()

Worker --->contains -many executor → to run→ task

One Partition ⇒ one task ⇒ one executor

RDD ⇒ Dataframe and Dataset

Data ⇒ metadata +data ⇒ table structure

To read from the CSV file

1. Header
2. inferSchema
3. Sep /delimiter

empdf.show(n)

empdf.limit(n).show()

Empdf.head(n)

Empdf.first()

empdf.tail()

empdf.rdd.getNumPartitions()

Distinct vs dropDuplicates(n)

Column filtering vs row filtering

Select filter /where

Read superstore.csv file as salesdf

Check the count and no of partition