Spark

1. Most important topic
2. Restart
3. Interview ✓
   + work

~1 month work + 1 year

(Storage) HDFS → distributed storage, though cloud based gaining property

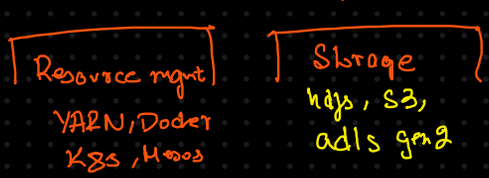
(resource manager) YARN → resource manager (Docker, Kubern, Mesos).

(Processing) MR → writing & performance ⟶ Spark

Some Common Questions.

Q:- Do I need to know hadoop before spark? → No

Q:- Do I need to install hadoop first? → No, Standalone



Q: Prerequisites before learning spark

→ Python, Java, Scala
→ SQL
→ hadoop essentials ⇒ distr. storage & processing

Q: Can you run spark on your laptop? ⇒ Yes

Q: Programming language?

Q: How is spark diff. from other databases? ⇒ Spark is not a dB

Q: What storage can you use with spark? ⇒ hdfs, s3, adls gen2, Kafka

Q: is spark free? ⇒ Yes

Android ⟶ OneUI

Q: What is diff b/w spark & pyspark?

Q: Can you use spark on cloud?

MR and its limitation

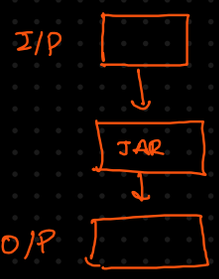
1. Performance → high latency
   disk i/o

2. Complex & hard to write :- hard to write boilerplate code
   verbose

3. Only batch processing is supported :- not stream processing

4. Rigid Sequence :- Everything has to be thought of interms of MR

5. No interactive mode/way to monitor jobs or run in ports?

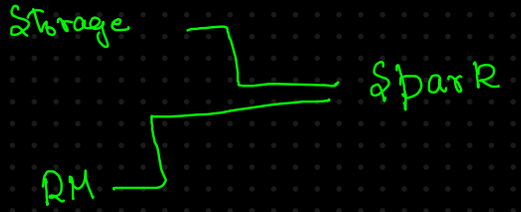


Spark

Spark 3

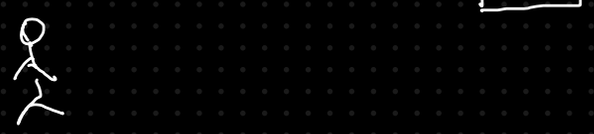
Apache Spark is an open-source distributed computing system designed to process large datasets quickly.

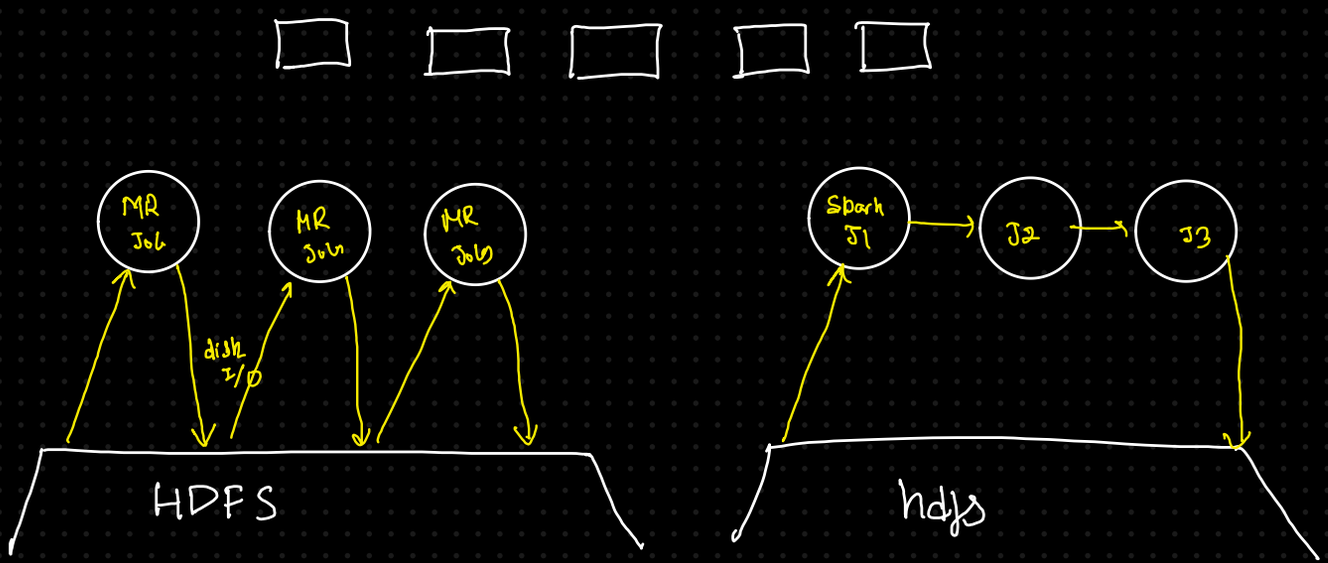
fast, general purpose & support batch & real time processing



Characteristic

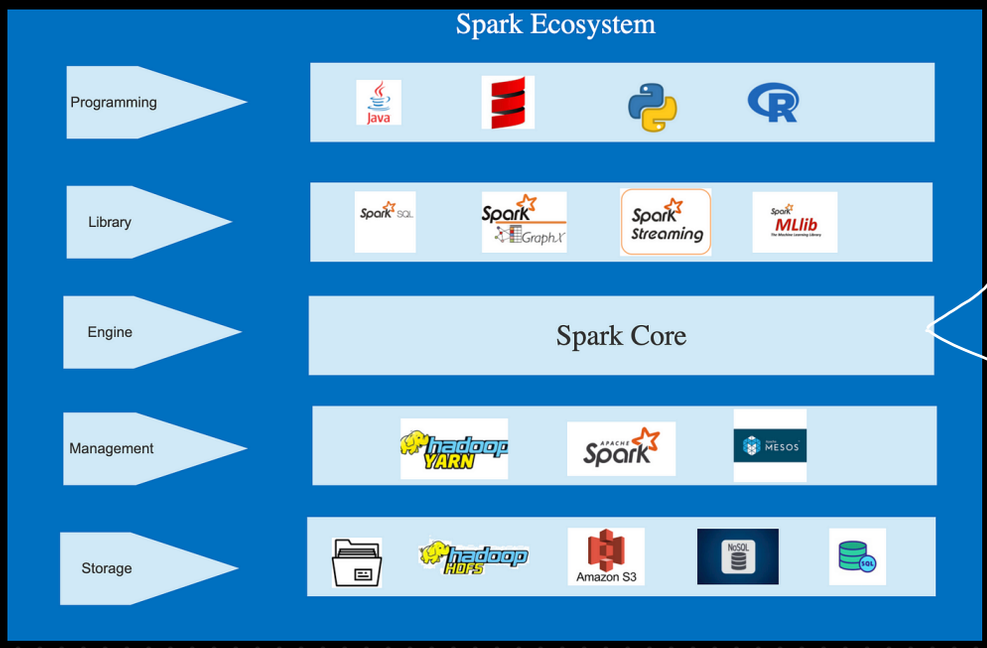
1. In-memory Processing: in-memory processing faster than traditional disk-based system





2. Ease of use → lang. support + lib

3. Unified Framework →



developed to make life easy (least flexible)

↑

Spark higher level

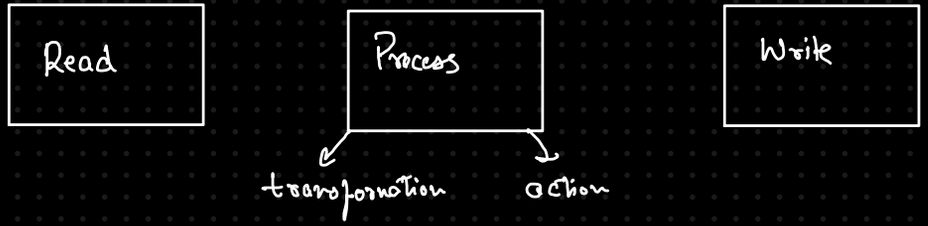
Spark core / low level

RDD (most flexible)

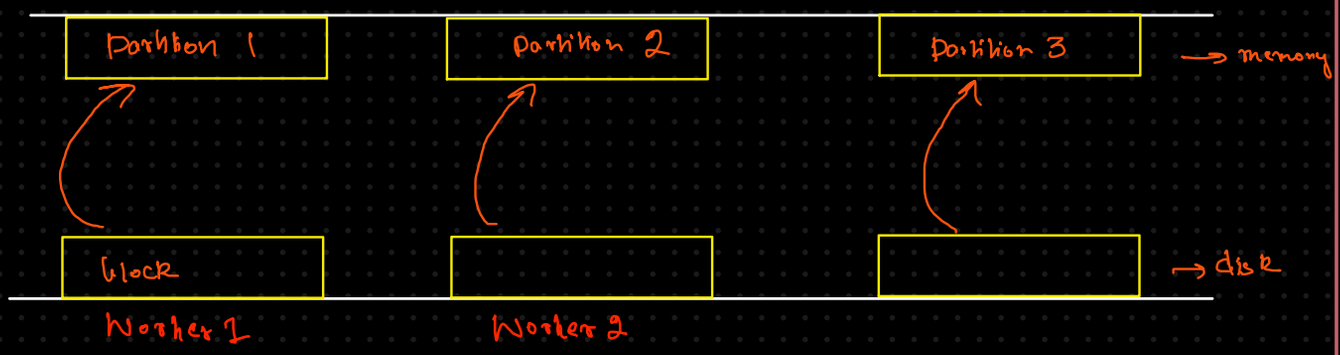
↓

Features of Apache spark

1. Speed : in memory processing, DAG,
2. Scalable : one m/c → 1000 m/c
3. Fault tolerant : automatically recovers the data in case of failure
4. Multi-language / Polyglot : multi lang support
5. Unified framework
6. Integration with existing systems : hdfs, yarn, Kafka



RDD & execution plan in Spark



data loaded from disk → memory

300mb ⇒ 3 blocks

block on disk
&
partition on memory

hdfs file → read in spark

↓

processing → Send it back to disk

rdd from some local data

rdd1 → → → →

filter, map...

↙ ↘

initialized data | hdfs

Nothing happened in our code unless we ran collect()

Lazy evaluation

Transformation

Opⁿ that create a new RDD from existing one.

Lazy

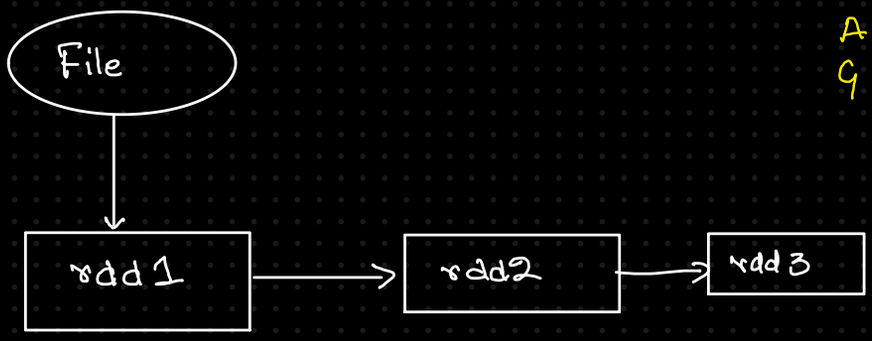
Action

Opⁿ that triggers the execution of all transformations & return result.

Eager

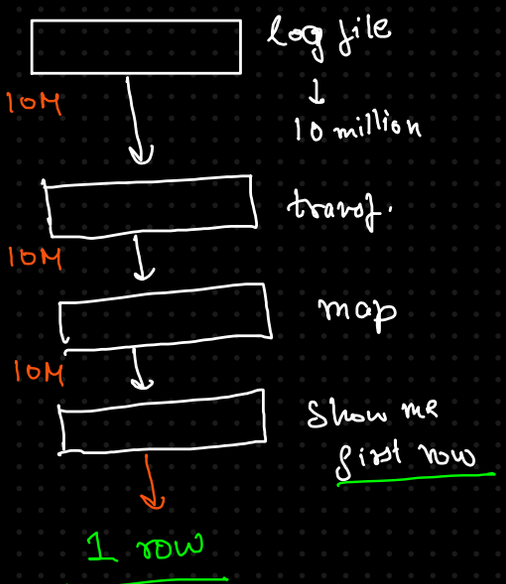
Cake

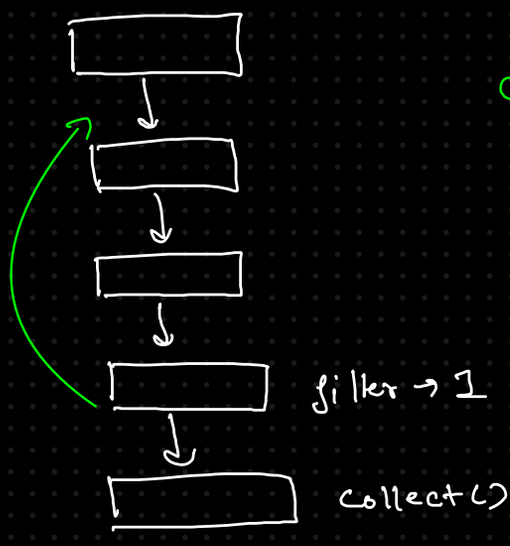
DAG / execution plan



D → Direct
A → Acyclic
G → Graph

Q: Why are transformation is lazy





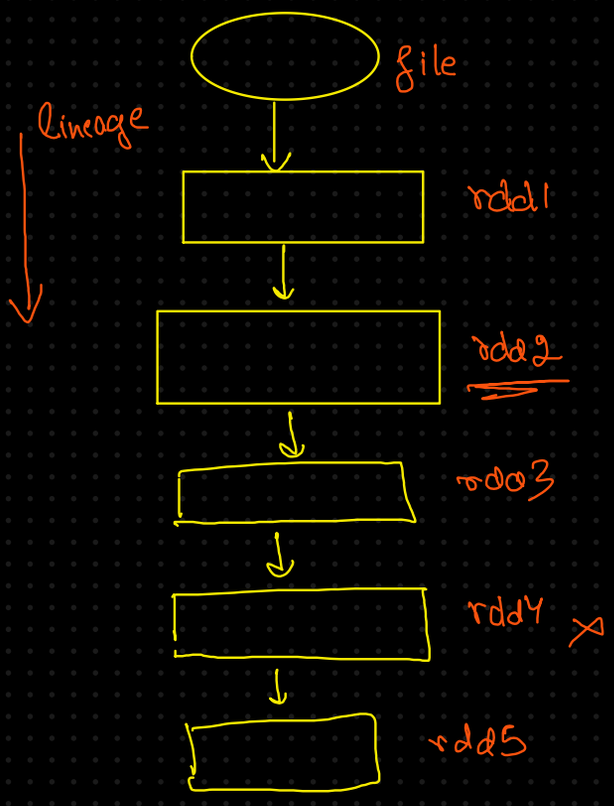
Transformations				
map	join	union	distinct	repartition
mapPartitions	flatMap	intersection	pipe	coalesce
cartesian	cogroup	filter	sample	
sortByKey	groupByKey	reduceByKey	aggregateByKey	
mapPartitionswithIndex		repartitionAndSortWithinPartitions		
Actions				
reduce	take	collect	takeSample	count
takeOrdered	countByKey	first	foreach	saveAsTextFile
saveAsSequenceFile		saveAsObjectFile		

R - resilient

D - distributed

D - dataset

Resilient will mean that we can quickly recover



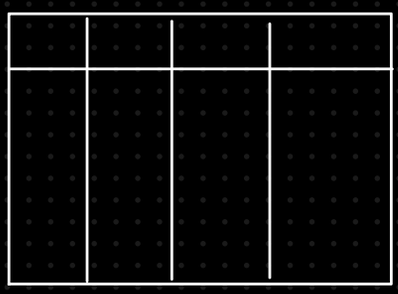
→ rdd are immutable

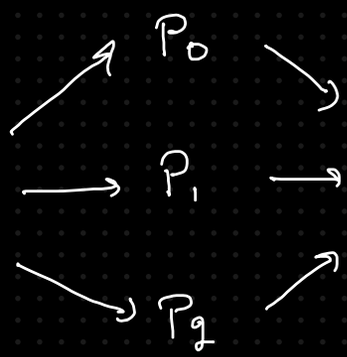
Properties of rdd

1. Immutable
2. Lazy evaluated
3. Partitioned

Characteristic

1. Foundamental data structure
2. Resilient
3. Fault Tolerance
4. Lazily evaluated





1. **SparkContext (sc)**:
   - It's the older entry point for Spark functionality
   - Primarily works with RDDs (Resilient Distributed Datasets)
   - Each JVM can only have one active SparkContext
   - More low-level control over Spark configurations
2. **SparkSession (spark)**:
   - Introduced in Spark 2.0 as the unified entry point
   - Encapsulates SparkContext, SQLContext, and HiveContext
   - Provides access to DataFrame and Dataset APIs
   - Can have multiple SparkSessions in the same application
   - Recommended for modern Spark applications
3. **"Normal" Spark**:
   - This usually refers to running Spark without specialized contexts
   - Basic functionality without SQL or Hive support
   - Limited compared to using SparkSession

Spark 2.0