Apache Spark Fundamentals

Eren Avşaroğulları

Data Science and Engineering Club Meetup

Dublin - December 9, 2017

Agenda

- What is Apache Spark?
- Spark Ecosystem & Terminology
- RDDs & Operation Types (Transformations & Actions) ★
- RDD Lineage
- Job Lifecycle ★
- RDD Evolution (DataFrames and DataSets) ★★
- Persistency ★
- Clustering / Spark on YARN

★ ***shows code samples***

Bio

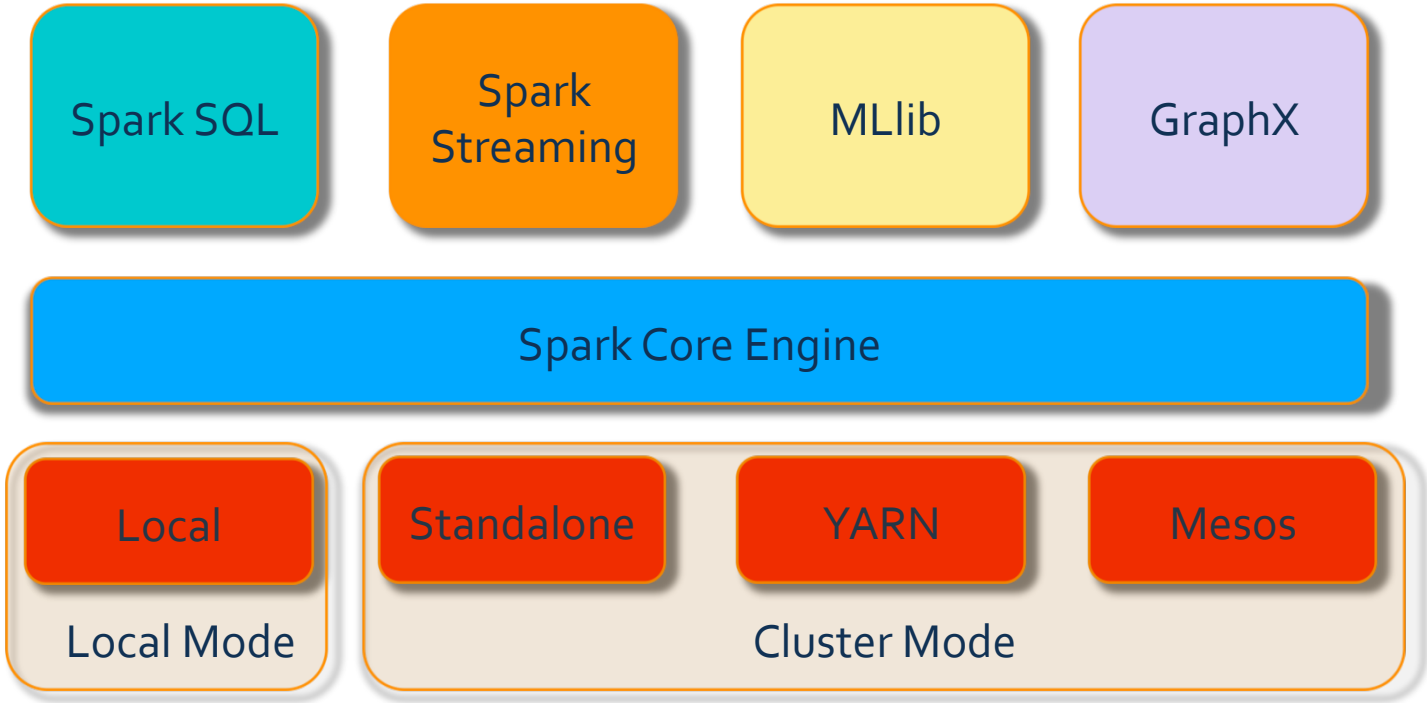
- B.Sc & M.Sc. on Electronics & Control Engineering
- Apache Spark Contributor since v2.0.0
- Sr. Software Engineer @ workday.
- Currently, work on Data Analytics
  Data Transformations / Cleaning

erenavsarogullari

What is Apache Spark?

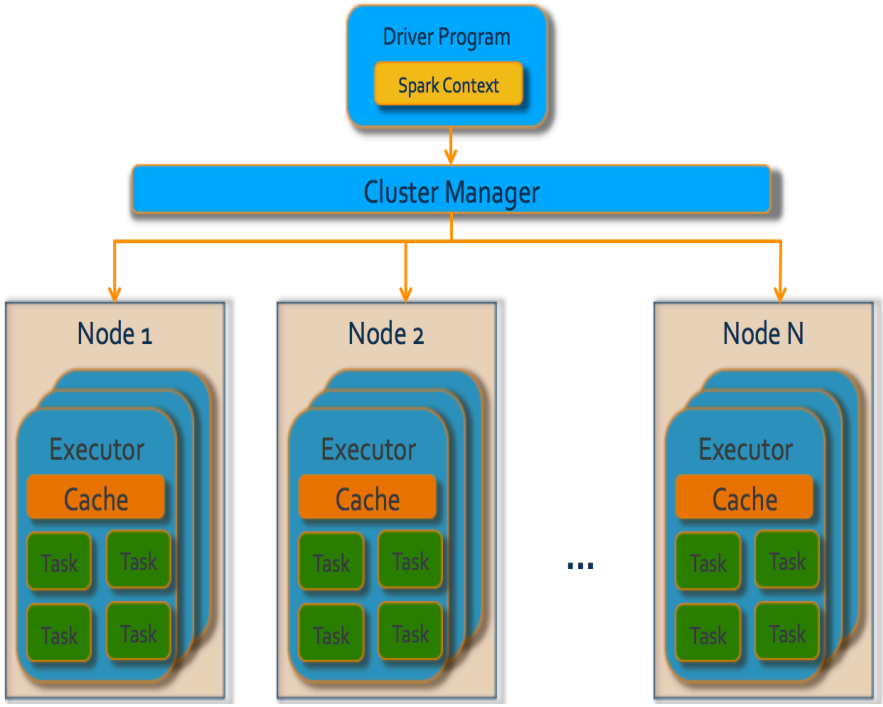
- Distributed Compute Engine
- Project started in 2009 at UC Berkley
- First version(v0.5) is released on June 2012
- Moved to Apache Software Foundation in 2013
- **Supported Languages:** Java, Scala, Python and R
- + 1100 contributors / +14K forks on Github
- **spark-packages.org** => ~380 Extensions

Spark Ecosystem



Terminology

- **RDD:** Resilient Distributed Dataset, immutable, resilient and partitioned.
- **DAG:** Direct Acyclic Graph. An execution plan of a job (a.k.a RDD dependency graph)
- **Application:** An instance of Spark Context. Single per JVM.
- **Job:** An **action** operator triggering computation.
- **Driver:** The program/process for running the Job over the Spark Engine
- **Executor:** The process executing a task
- **Worker:** The node running executors.



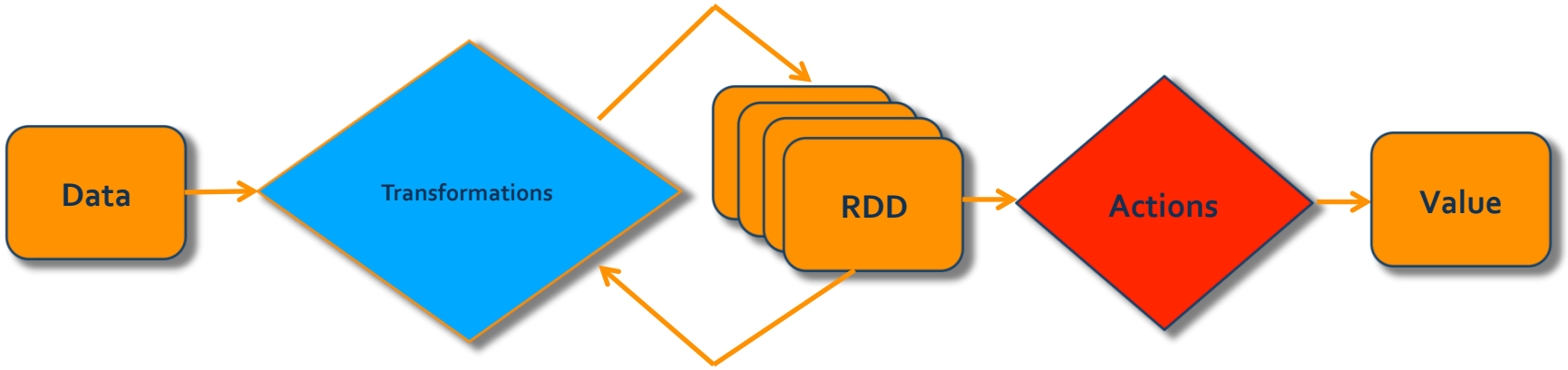
How to create RDD?

- Collection Parallelize
- By Loading file
- Transformations
- **Lets see the sample =>** Application-1 

RDD Operation Types

Two types of Spark operations on RDD

- **Transformations**: lazy evaluated (not computed immediately)
- **Actions:** triggers the computation and returns value



Transformations

- **map**(func)
- **flatMap**(func)
- **filter**(func)
- **union**(dataset)
- **join**(dataset, usingColumns: Seq[String])
- **intersect**(dataset)
- **coalesce**(numPartitions)
- **repartition**(numPartitions)

Full List:

https://spark.apache.org/docs/latest/rdd-programming-guide.html#transformations

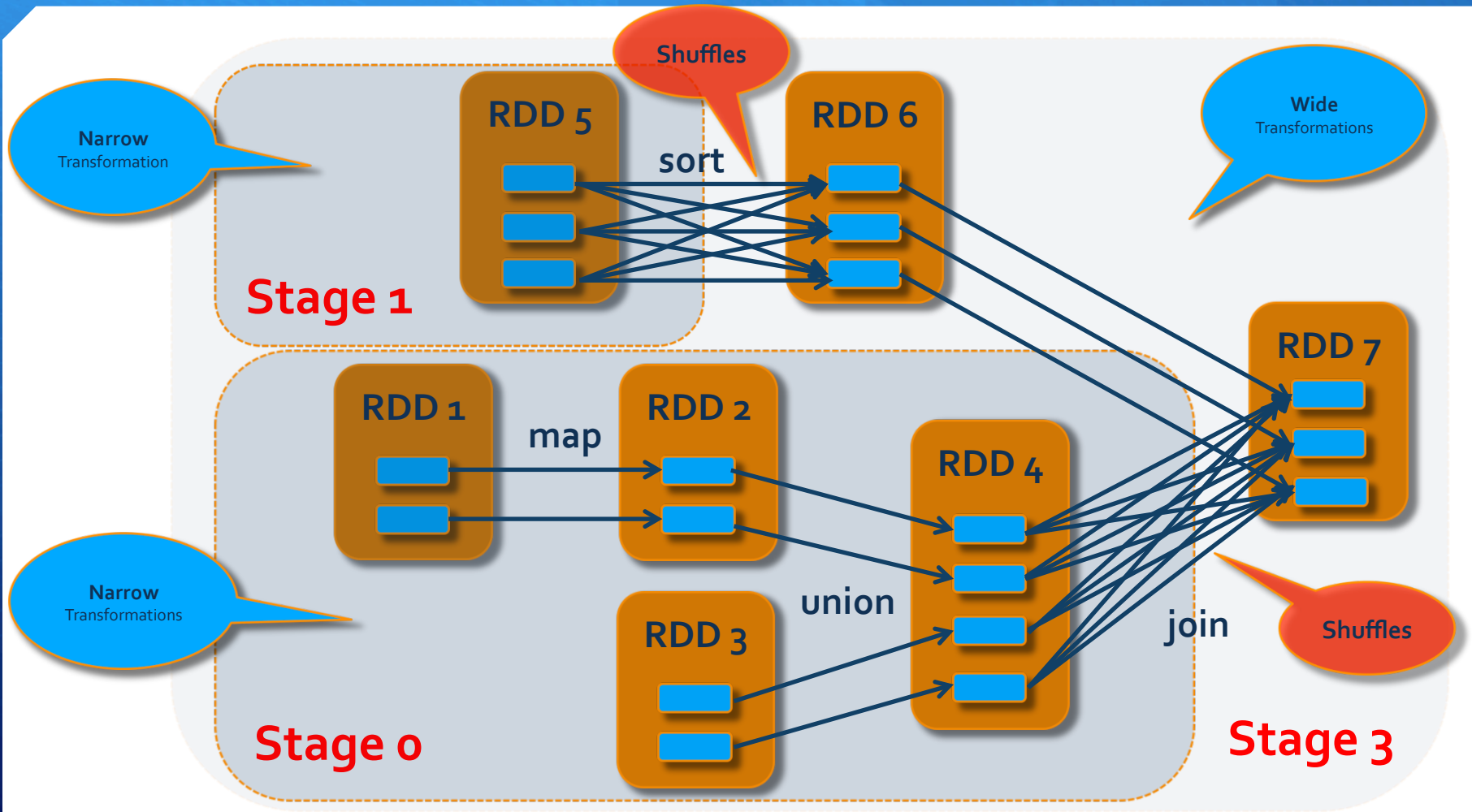
Actions

- first()
- take(n)
- collect()
- count()
- saveAsTextFile(path)

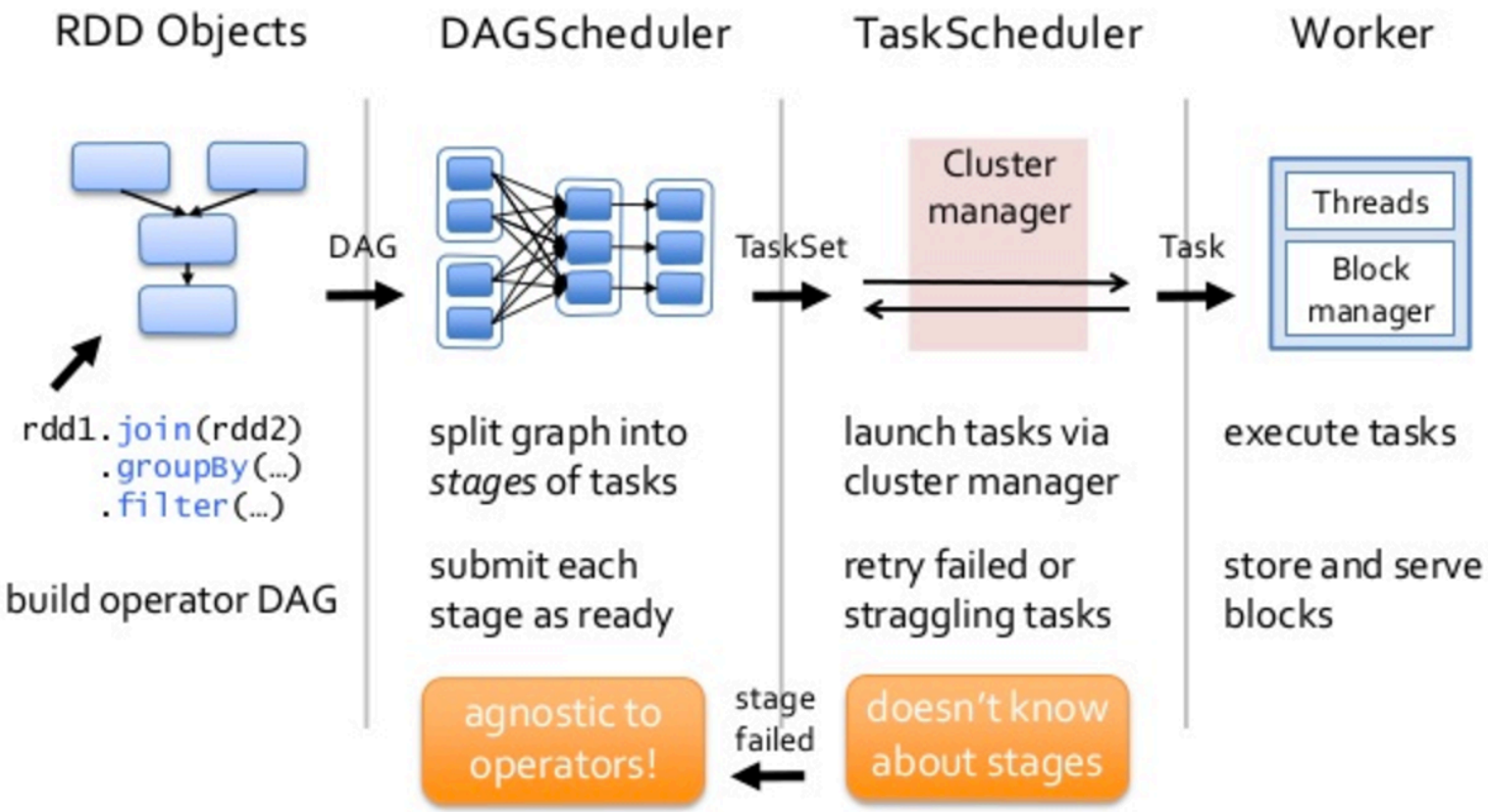
Full List:
https://spark.apache.org/docs/latest/rdd-programming-guide.html#actions

Lets see the sample => Application-2 

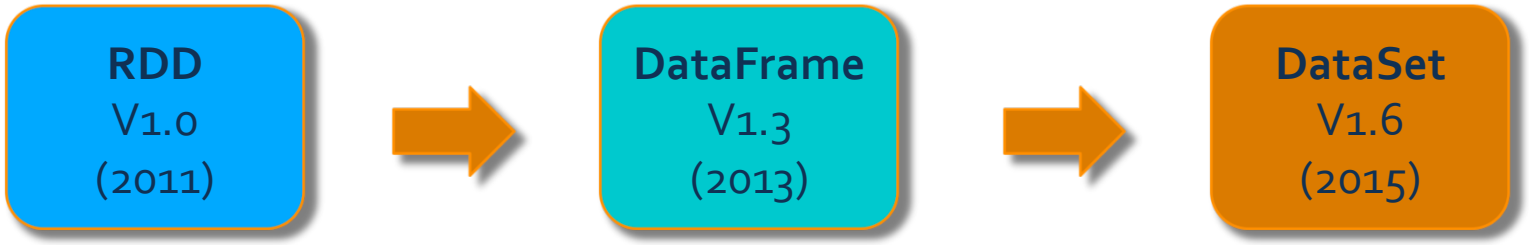
RDD Dependencies (Lineage)



Job Lifecyle



RDD Evolution



Java Objects
Low level data-structure
To work with
Unstructured Data

Untyped API
Schema based - Tabular

Typed API: [T]
Tabular

SQL Support
To work with
Semi-Structured (csv, json) / Structured Data (jdbc)
Project Tungsten
Catalyst Optimizer

Two tier optimizations

How to create the DataFrame?

- By loading file (spark.read.format("csv").load())
- SparkSession.createDataFrame(**RDD**, **schema**)

Lets see the code – Application-3 

How to create the DataSet?

- By loading file (spark.read.format("csv").load())
- SparkSession.createDataSet(**collection** or **RDD**)

Lets see the code – Application-4-1 

Application-4-2 

Persistency

Storage Modes	Details
MEMORY_ONLY	Store RDD as deserialized Java objects in the JVM
MEMORY_AND_DISK	Store RDD as deserialized Java objects in the JVM
MEMORY_ONLY_SER	Store RDD as *serialized* Java objects (Kryo API can be thought)
MEMORY_AND_DISK_SER	Similar to MEMORY_ONLY_SER
DISK_ONLY	Store the RDD partitions only on disk.
MEMORY_ONLY_2, MEMORY_AND_DISK_2	Same as the levels above, but replicate each partition on two cluster nodes.

- RDD / DF.persist(newStorageLevel: StorageLevel)
- RDD.unpersist() => Unpersists RDD from memory and disk

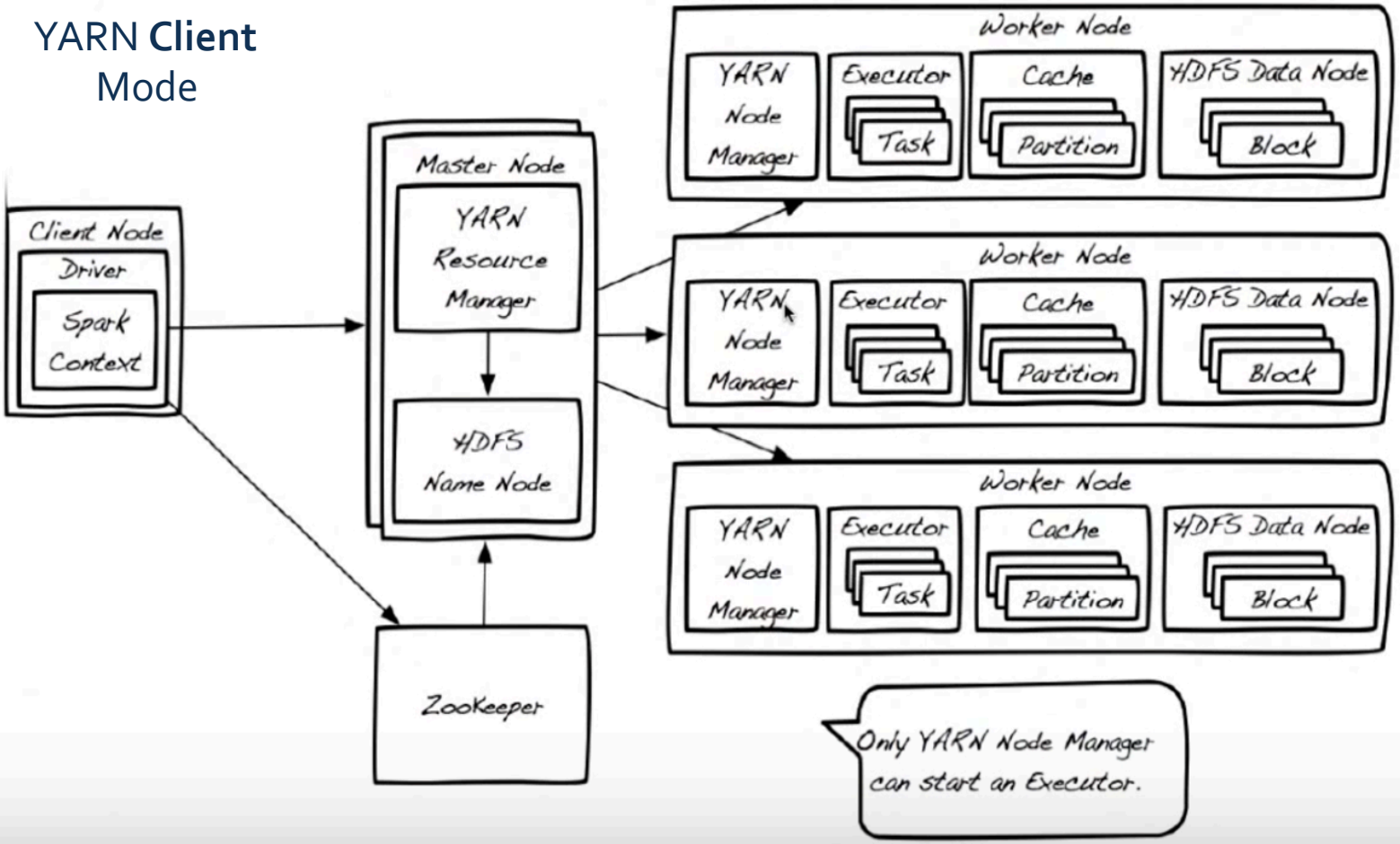
Unpersist will need to be **forced** for **long term** to use executor memory efficiently.

Note: Also when cached data exceeds storage memory,

Spark will use Least Recently Used(LRU) Expiry Policy as default

Clustering / Spark on YARN

YARN **Client** Mode



Q & A

Thanks

References

- https://spark.apache.org/docs/latest/
- https://cwiki.apache.org/confluence/display/SPARK/Spark+Internals
- https://jaceklaskowski.gitbooks.io/mastering-apache-spark
- https://stackoverflow.com/questions/36215672/spark-yarn-architecture
- High Performance Spark by

  Holden Karau & Rachel Warren