A gentle introduction to

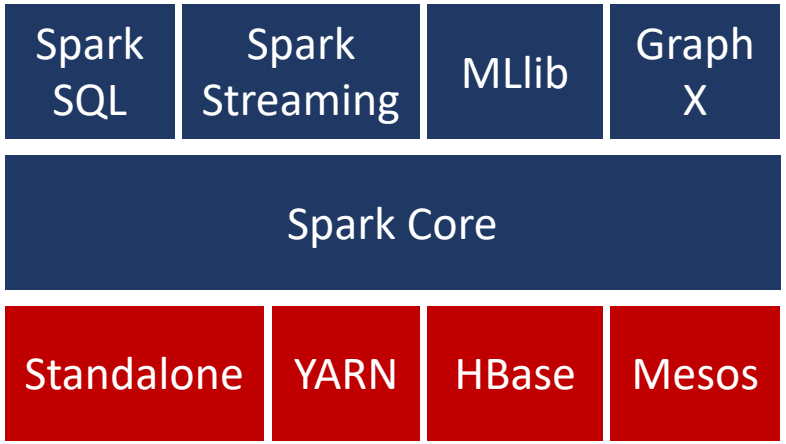
Alessandro Mele
Ph.D. student
a.mele@pm.univpm.it

Apache Spark

- Standard de facto per i Big Data Analytics
- Veloce
  - Lavora in-memory
  - 10 volte più di MapReduce
- Semplice da utilizzare

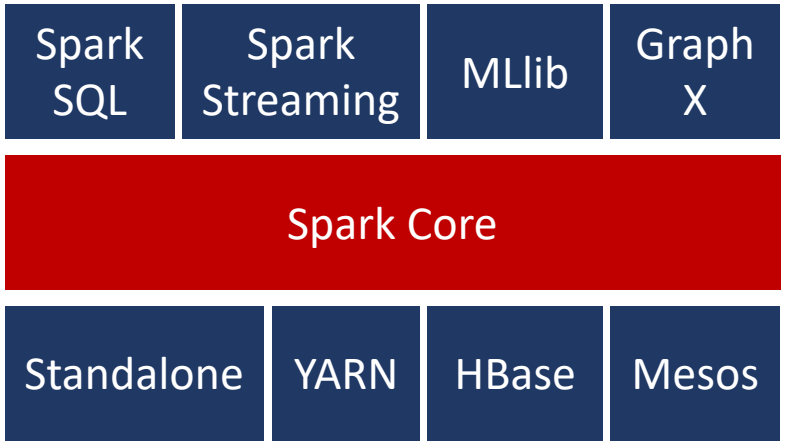
Spark: Stack

- **Cluster manager**: abilita l'accesso a risorse di calcolo distribuite
  - Standalone
  - YARN
  - Hbase
  - […]



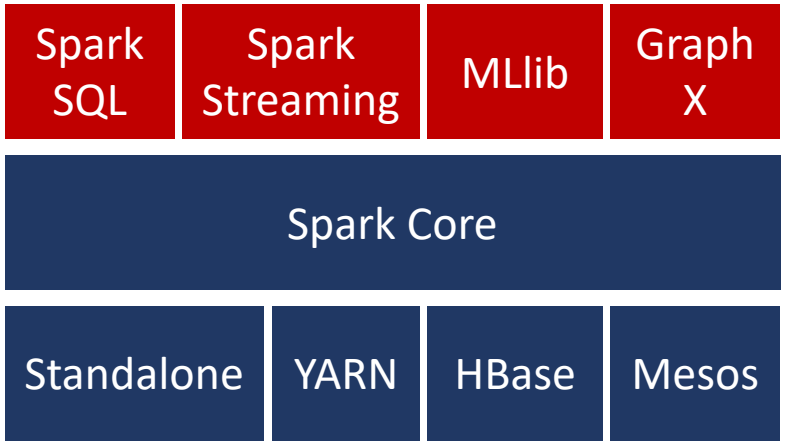
Spark: Stack

- **Spark core**: interfaccia di programmazione per l'elaborazione dei dati
- Fornisce API per Python, Scala, R, […]
  - *Transformation:* manipolazione *lazy* su RDD
  - *Action*: eseguono le *Transformation* e restituiscono il risultato dell'elaborazione



Spark: Stack

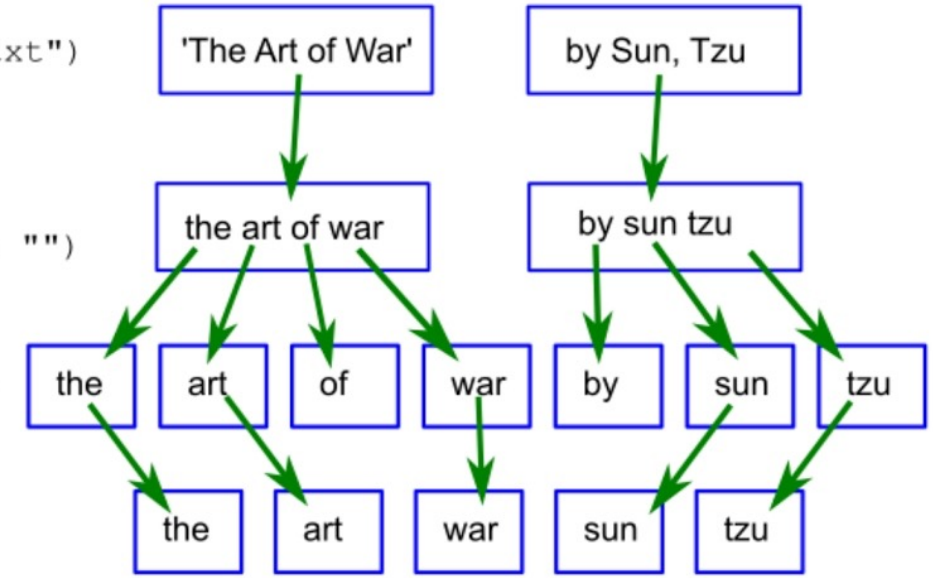
- **Librerie**: interfaccia di programmazione per l'elaborazione dei dati
  - Standard: Spark SQL, Spark Streaming, MLlib, GraphX
  - Terze parti: Koalas, Mlflow, GeoSpark, […]



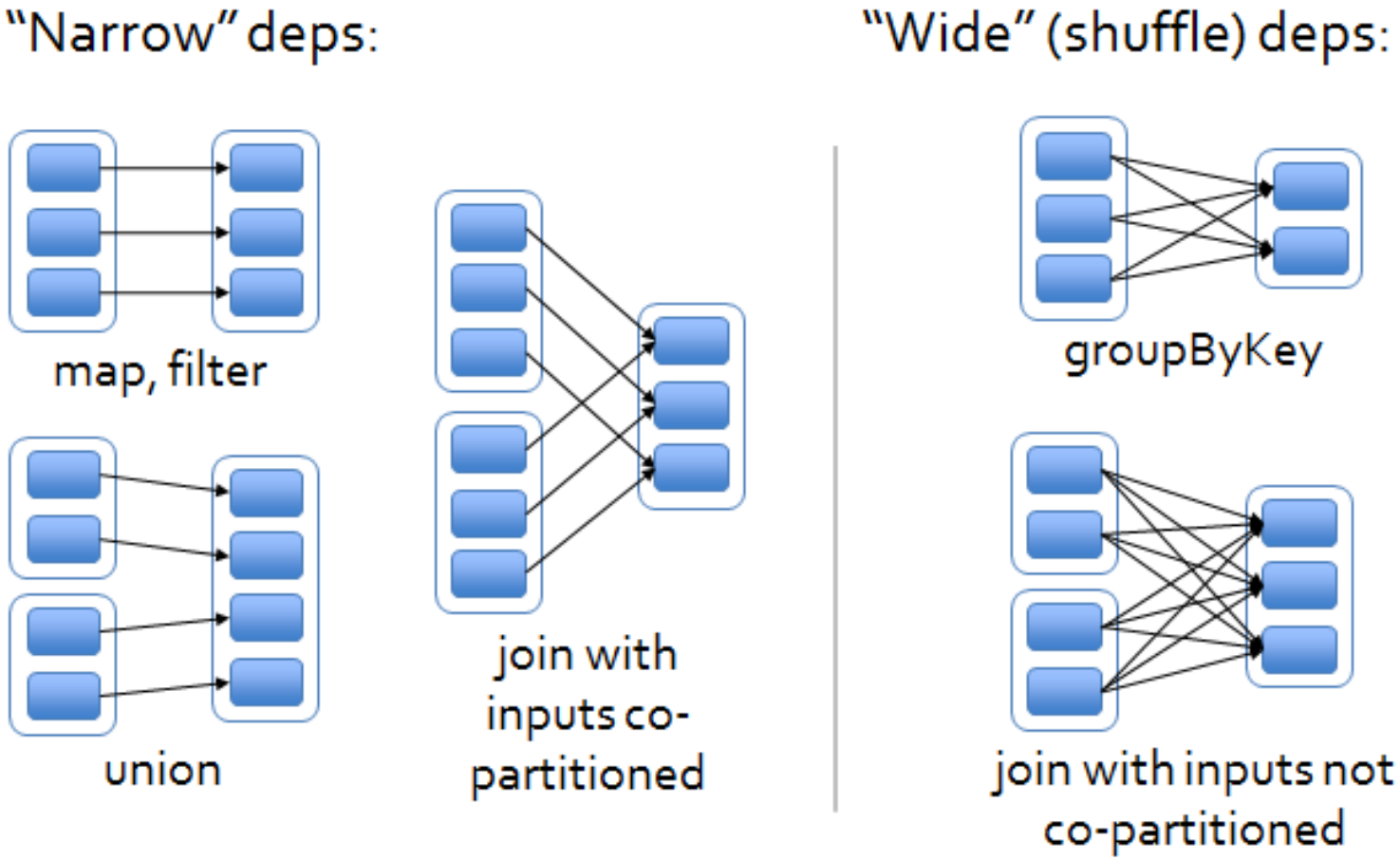
Spark: RDD

- Collezioni di dati partizionate e Read-only
- Si rappresenta con un DAG
- Resilienti: in caso di problemi, si riprende l'esecuzione esattamente dal punto dove è stato interrotto

```
sc.textFile("artofwar.txt")
.map(
  _.toLowerCase
  .replaceAll("[^\\w ], "")
.flatMap(_.split(" "))
.filter(_.size >2)
```



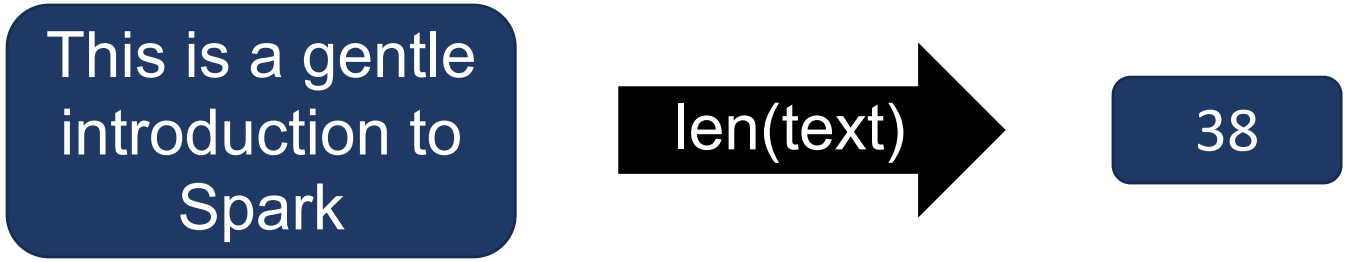
Spark: Transformation



Spark: Transformation

- **Map**

  Return a new distributed dataset formed by passing each element of the source through a function *func*



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

Spark: Transformation

- **FlatMap**

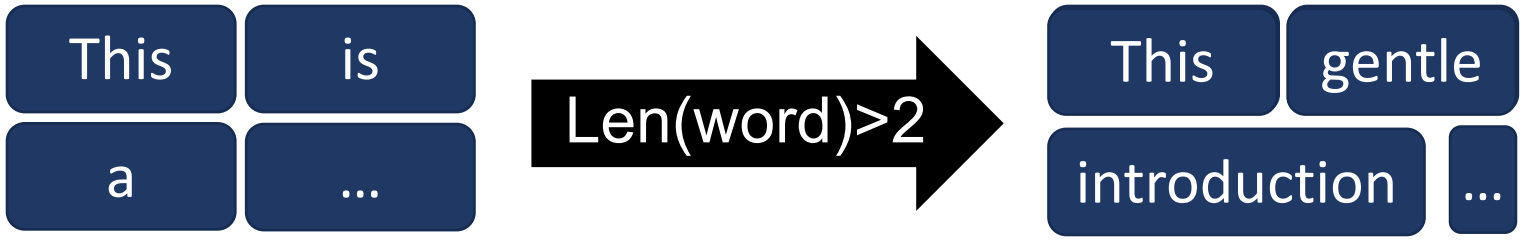
  Similar to map, but each input item can be mapped to 0 or more output items (so *func* should return a Seq rather than a single item)

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

Spark: Transformation

- **Filter**

  Return a new dataset formed by selecting those elements of the source on which *func* returns true

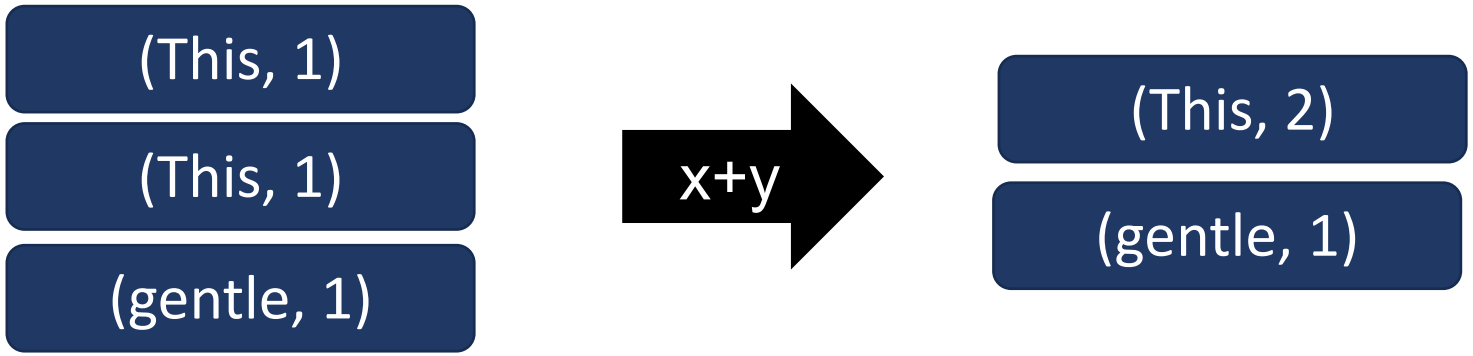


[1] https://spark.apache.org/docs/latest/rdd-programming-guide.html#transformations

Spark: Transformation

- **ReduceByKey**

  When called on a dataset of (K, V) pairs, returns a dataset of (K, V) pairs where the values for each key are aggregated using the given reduce function *func*, which must be of type (V,V) => V



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

Spark: Actions

- **Reduce**
  - Aggregate the elements of the dataset using a function *func* (which takes two arguments and returns one)
- **Collect**
  - Return all the elements of the dataset as an array at the driver program
- **Count**
  - Return the number of elements in the dataset

[2] https://spark.apache.org/docs/latest/rdd-programming-guide.html#actions

Spark: Actions

- **Take**
  - Return an array with the first *n* elements of the dataset.
  - **First** return the first element of the dataset
- **countByKey**
  - Only available on RDDs of type (K, V). Returns a hashmap of (K, Int) pairs with the count of each key
- **Foreach**
  - Run a function *func* on each element of the dataset

[2] https://spark.apache.org/docs/latest/rdd-programming-guide.html#actions

Spark: Note

- Dopo la lezione, sarà disponibile un repository GitHub contenente sia gli esercizi che le slide
  - [3] https://github.com/AlessandroMele/EsercitazioniBDA-ML

Let’s start coding!