Laboratorio 5. (3° giorno - mattino)

Obiettivi

- Fare qualche esperimento con le RDD.

Riferimenti

https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.rdd.RDD

Per creare un nuovo RDD:

```
val rdd1 = sc.parallelize(arr, 10)
val rdd2 = sc.textFile("/spark/divina_commedia.txt")
```

Dove "arr" è un array Scala e 10 (opzionale) è il numero di partizioni. N.B. anche nella *textFile* si può specificare il numero di partizioni.

Per vedere il contenuto si possono usare le funzioni *first, take, collect*. N.B. *show* non è definita per gli RDD.

Ad es. per vedere le prime righe della divina commedia si può scrivere:

```
rdd1.take(20).foreach(println)
```

Per manipolare gli rdd si possono usare le funzioni *map. flatMap, filter, ecc*.

Gli RDD di coppie accettano anche altri metodi: *groupBy, reduceBy*. ecc. (vedi: https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.rdd.PairRDDFunctions)

Esercizi

Esercizio.1

Leggere il file *spark/divina_commedia.txt* in un RDD. Stampare le prime 20 righe saltando le righe vuote. Stampare tutte le linee che contengono la parola *amor*.

Esercizio.2

Creare un RDD con un determinato numero di partizioni. Verificare il numero delle partizioni con *rdd.getNumPartitions* oppure *rdd.partitions.length*. Generate un nuovo RDD con un numero diverso di partizioni usando il comando *repartition(n)* e verificate il nuovo numero di partizioni.

Esercizio.3

Usate il metodo glom per verificare, nell'esercizio precedente, come i dati vengono distribuiti nelle varie partizioni.

Esercizio.4

Create un RDD che contiene un elenco di parole con ripetizioni e fate un conteggio delle parole: ad esempio *("a","b","a")* deve diventare *(("a",2),("b",1))*.