Spark

• The paradigm of Map-reduce is capable of handling big data by the concept of divide and conquer.

• However, in the reduce part, its requirement to collect multiple computational results in a single point takes time.

• If an algorithm, needs to be run iteratively, a mapreduce job is very time consuming.

• So Hadoop needs Spark!

```
line_rdd = sc.textFile('shake.txt')
line_read.take(2)
line_rdd.count()
words_rdd = line_rdd.flatMap(lambda x:x.split())
words_rdd.take(2)
words_rdd.count()
words_rdd.distinct().count()
```

Spark MapReduce

```
#Lazy Programming
#
```

• Spark is developed in UC Berkley AMPLab by Matei Zaharia in 2009.

• Spark is written in Scala (Most other Hadoop apps were written in Java).

• Spark can handle both real time and batch computational tasks on HDFS and other data sources.

• Spark is an in-memory computational framework so that if a data set can fit into a memory, the computation is much faster (100x).

• Map reduce needs to be done into small pieces in many applications but Spark can be implemented continuously in a sequence of in-memory data.

• Spark performs better for iterative algorithms.RDD (Resilient Distributed Datasets): a programming abstraction that represents a read-only collection of object that are partitioned across a set of machines.

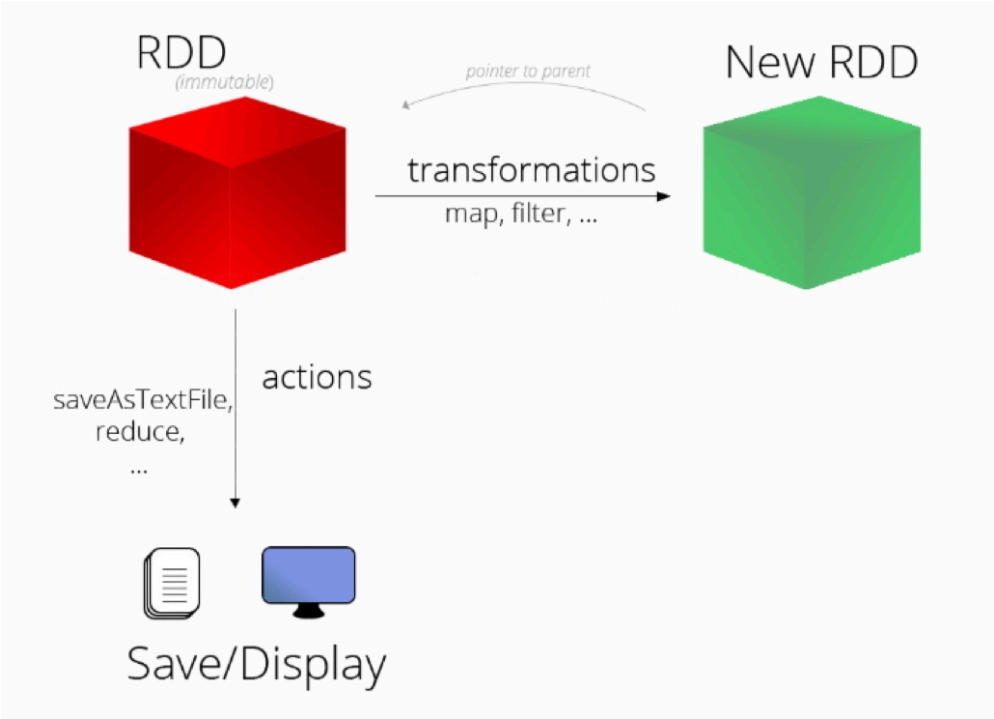
• Two types of operations on RDDs:

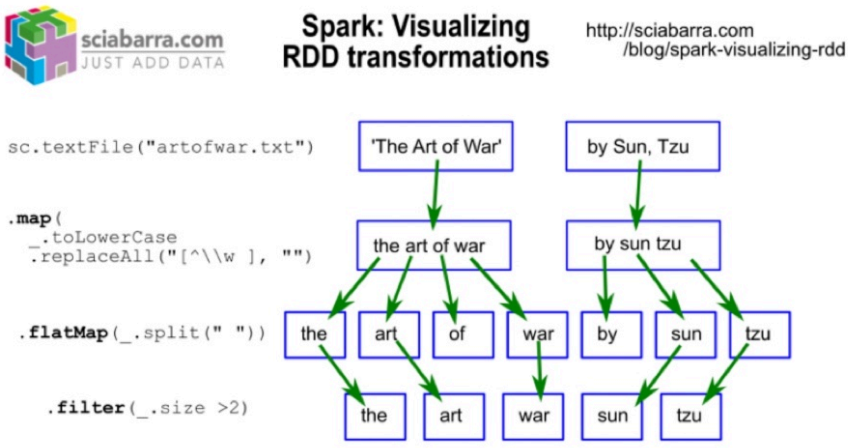
– Transformations (like map)

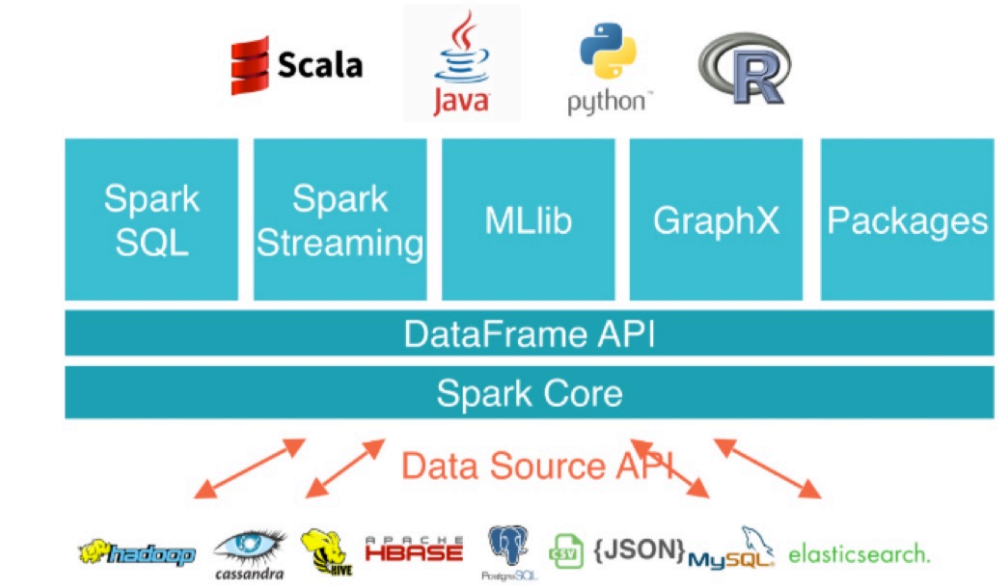
– Actions (like reduce) e.g. sum or mean

• RDDs are partitioned collections of data that allow the programmer to apply operations to the entire collection

in parallel.


RDD
(immutable)
pointer to parent
New RDD
transformations
map, filter, ...
saveAsTextFile,
reduce,
...
actions
Save/Display


sciabarra.com
JUST ADD DATA
Spark: Visualizing
RDD transformations
http://sciabarra.com
/blog/spark-visualizing-rdd
sc.textFile("artofwar.txt")
.map(
_.toLowerCase
.replaceAll("[^\\w], "")
.flatMap(_.split(" "))
.filter(_.size >2)
'The Art of War'
by Sun, Tzu
the art of war
by sun tzu
the
art
of
war
by
sun
tzu
the
art
war
sun
tzu



If you are running in stand_alone mode, please have these two rows first:

– conf = SparkConf().setMaster("local").setAppName("WordCount")

– sc = SparkContext(conf = conf)

```
lines_rdd.count()
words_rdd=lines_rdd.flatMap(lambda x: x.split())
words_rdd.count()
words_rdd.distinct().count()
words_rdd.countByValue()
#First assign 1 as value to each key:
key_value_rdd=words_rdd.map(lambda x: (x,1))
key_value_rdd.take(5)
word_counts_rdd=key_value_rdd.reduceByKey(lam bda x,y:x+y)
word_counts_rdd.take(5)
flipped_rdd=word_counts_rdd.map(lambda x: (x[1], x[0]))
flipped_rdd.take(5)
results_rdd=flipped_rdd.sortByKey(False)
results_rdd.take(5)
#Combine all these codes into one
lines_rdd.flatMap(lambda x: x.split()).map(lambda x:\
(x,1)).reduceByKey(lambda x,y:x+y).map(lambda x: (x[1],x[0])).sortByKey(False).take(5)

```