



Apache Spark

Χρήστος Γκόγκος

24/1/2022

https://github.com/chgogos/big_data



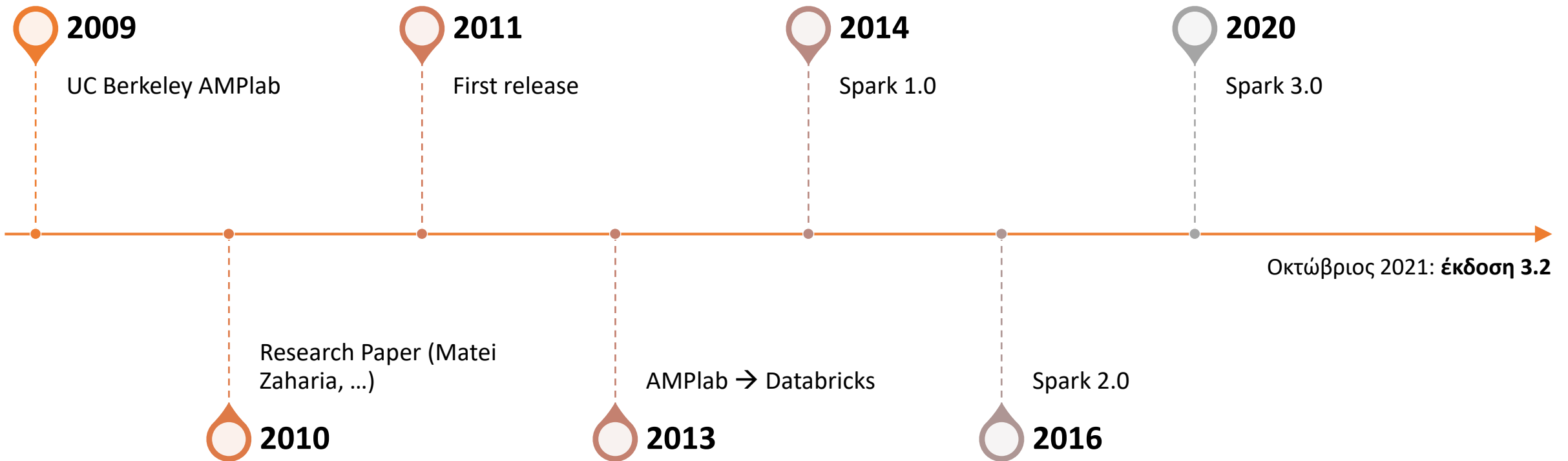
Τι είναι το Apache Spark;

Το Apache Spark είναι ένα framework γενικού σκοπού που επιτρέπει κατανεμημένη επεξεργασία σε ομάδες υπολογιστών

Μπορεί να θεωρηθεί ως μηχανή επεξεργασίας (processing engine) που δίνει έμφαση σε ταχύτητα και ευκολία χρήσης ενώ παράλληλα παρέχει δυνατότητες προχωρημένης ανάλυσης δεδομένων

Εφόσον είναι δυνατό διατηρεί τα δεδομένα στη κύρια μνήμη των executors (2x έως 100x ταχύτερο από το Hadoop MapReduce για συγκεκριμένες εργασίες)

Ιστορία του Apache Spark





Apache Spark

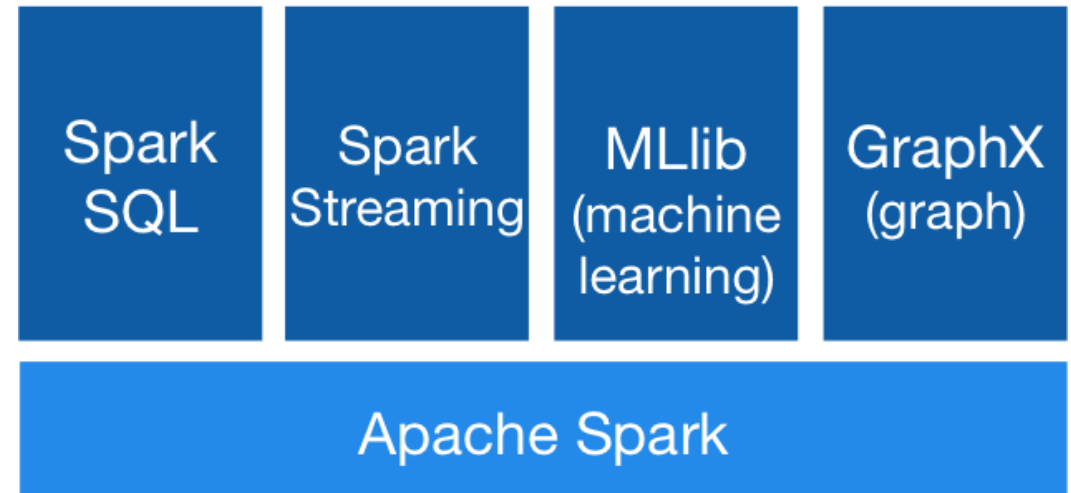
“A unified computing engine and a set of libraries for big data”

- **Unified:** Επιτρέπει τη φόρτωση και επεξεργασία δεδομένων με ενοποιημένο API, εύκολο στην κατανόηση (π.χ. φόρτωση δεδομένων χρησιμοποιώντας ένα SQL ερώτημα → είσοδος δεδομένων σε μοντέλο μηχανικής μάθησης → αποθήκευση αποτελεσμάτων σε επιθυμητή μορφή)
- **Computing Engine:** Το Apache Spark εστιάζει στη φόρτωση δεδομένων από συστήματα αποθήκευσης και στην εκτέλεση υπολογισμών στα δεδομένα όπου αυτά βρίσκονται (δεν αποτελεί το ίδιο λογισμικό αποθήκευσης δεδομένων)
- **Libraries:** Παρέχει API που εξυπηρετεί συχνές εργασίες ανάλυσης δεδομένων (περιέχει standard libraries + 3rd party libraries)

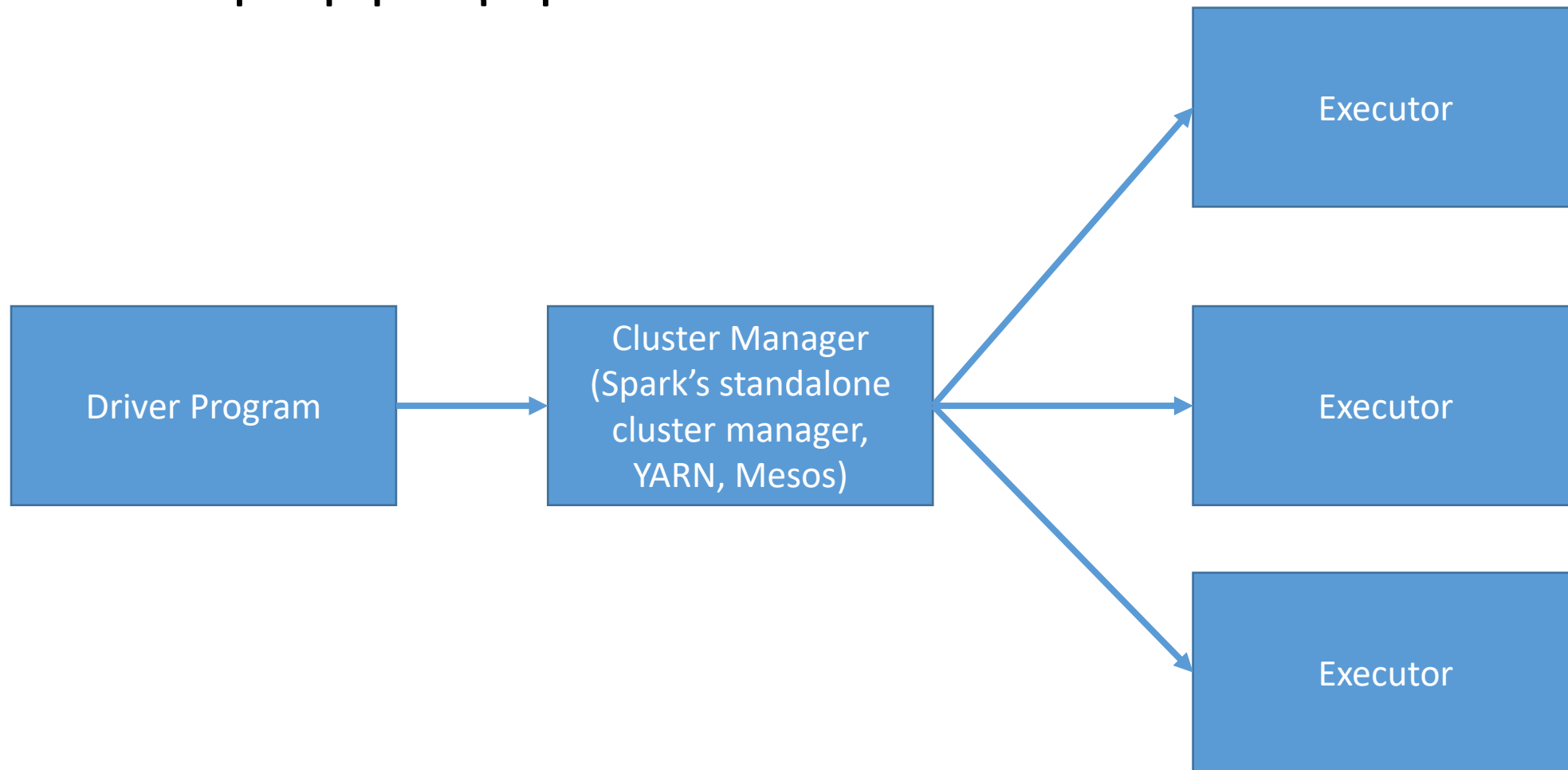
Το Stack του Apache Spark



- **Spark SQL:** πρόσβαση σε δομημένα δεδομένα – συμβατότητα με HiveQL
- **Spark Streaming:** fault tolerant χειρισμός data streams (Flume, Kafka,...)
- **MLlib:** έλεγχος υποθέσεων, κατηγοριοποίηση, παλινδρόμηση, συσταδοποίηση, ανάλυση κύριων συνιστωσών κ.α.
- **GraphX:** ανάλυση γραφημάτων (π.χ. pagerank), μέσω του Pregel API



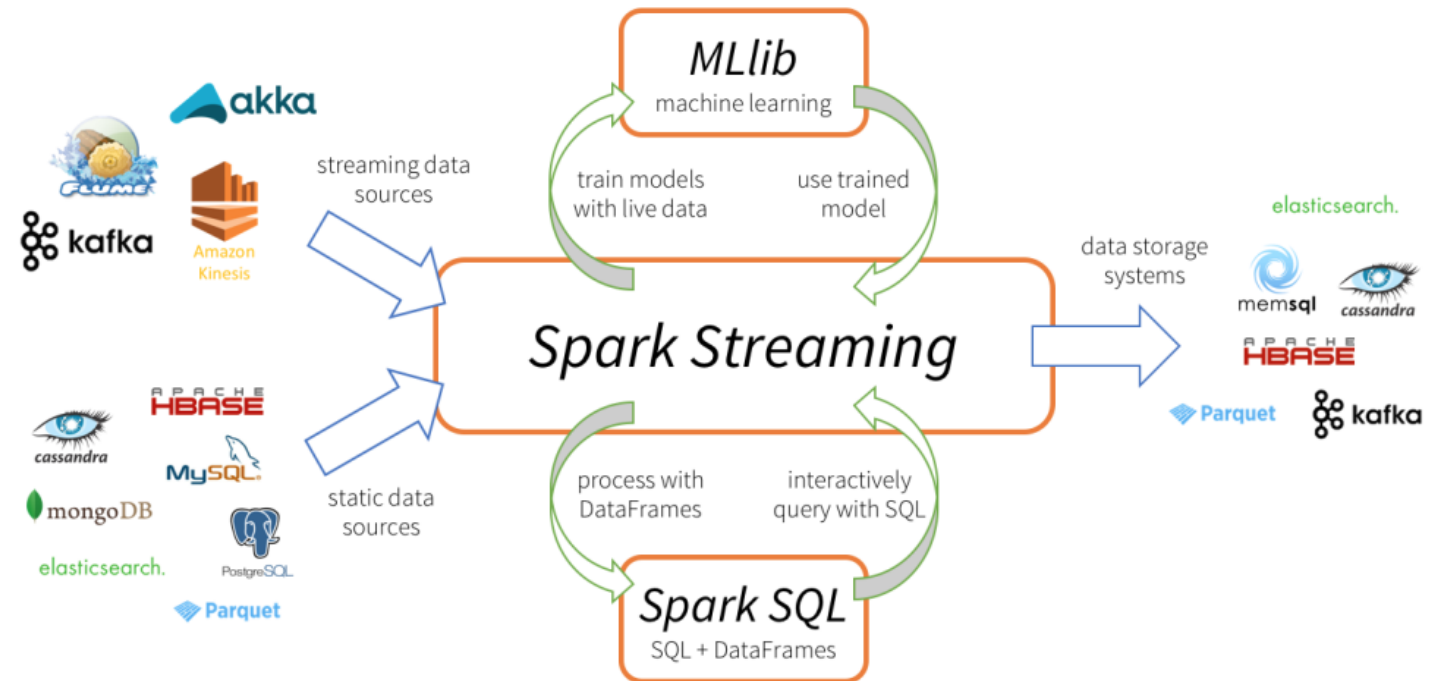
Spark εφαρμογή



Για ποιες εφαρμογές
είναι κατάλληλο το
Apache Spark;

Το Spark είναι κατάλληλο για:

- Διαδραστικά ερωτήματα σε μεγάλα δεδομένα
- Επεξεργασία streaming μεγάλων δεδομένων από αισθητήρες ή από άλλες πηγές
- Ανάπτυξη εφαρμογών μηχανικής μάθησης σε μεγάλα δεδομένα



<https://databricks.com/blog/2016/06/22/apache-spark-key-terms-explained.html>

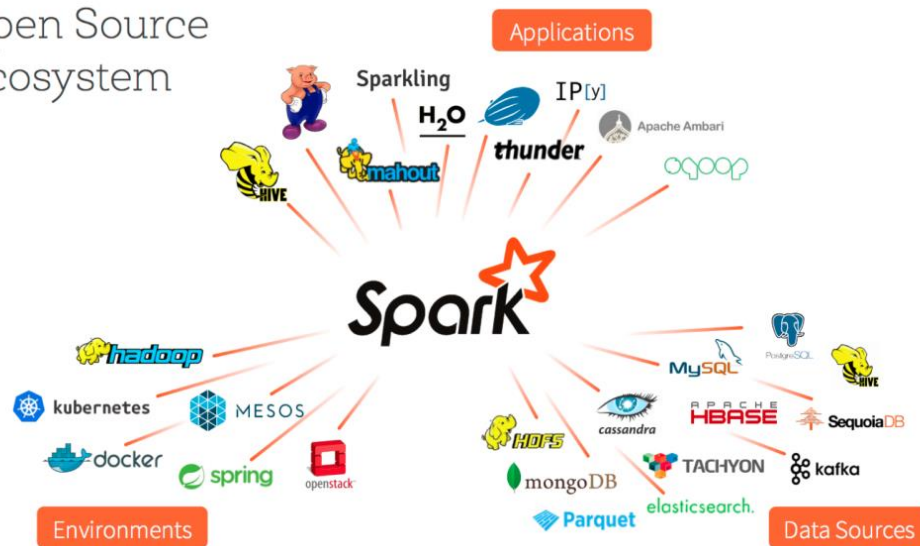
Χαρακτηριστικά του Spark

- Διατηρεί τα δεδομένα και τα ενδιάμεσα αποτελέσματα στη μνήμη, αντί να τα γράφει στο δίσκο
 - Παρέχει επεξεργασία «σχεδόν» πραγματικού χρόνου
 - Σε σχέση με το Hadoop MapReduce το Spark:
 - πραγματοποιεί λιγότερο «ακριβά» ανακατέματα (shuffles) κατά την επεξεργασία δεδομένων
 - Παρέχει υψηλότερου επιπέδου API που διευκολύνει τους προγραμματιστές
- Έχει σχεδιαστεί ως **μηχανισμός εκτέλεσης εφαρμογών τόσο στη μνήμη όσο και στο δίσκο** (όταν η μνήμη δεν επαρκεί, οι λειτουργίες του Spark χρησιμοποιούν διαθέσιμες δευτερεύουσες μονάδες αποθήκευσης)

Δεδομένα που χειρίζεται το Spark

- Συχνά χρησιμοποιείται πάνω από το Hadoop που του παρέχει πρόσβαση σε δεδομένα τα οποία βρίσκονται στο HDFS ή στην HBase

Open Source
Ecosystem



- Επιπλέον:

- Μπορεί να διαβάσει δεδομένα και από άλλα συστήματα αποθήκευσης δεδομένων όπως Cassandra, MongoDB, CouchDB κ.α.
- Μέσω του υποσυστήματος Apache Spark SQL μπορεί να έχει πρόσβαση μέσω SQL σε σχεσιακές βάσεις δεδομένων
- Μπορεί να χρησιμοποιεί το Apache Mesos ως cluster manager και να εκτελείται εκτός Hadoop σε ομάδες υπολογιστών που τη διαχείρισή τους αναλαμβάνει το Mesos



Γλώσσες που υποστηρίζει το Spark


- Scala
 - Java
 - Python
 - R
- Διαθέτει REPL (Read Evaluate Print Loop) για: Scala, Python, R
 - Python notebooks
 - R notebooks

```
$ spark-shell  
20/05/17 21:04:24 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-  
java classes where applicable  
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
Spark context Web UI available at http://DESKTOP-66P02VI:4040  
Spark context available as 'sc' (master = local[*], app id = local-1589738670215).  
Spark session available as 'spark'.  
Welcome to
```

```
version 2.4.5
```

```
Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_181)  
Type in expressions to have them evaluated.  
Type :help for more information.
```

```
scala> 
```



The screenshot shows a Jupyter Notebook window titled "Untitled5" with a last checkpoint of 4 minutes ago. The interface includes a top menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with various icons. The main area contains a code cell with the following Python code:

```
In [11]: import findspark
findspark.init()

import pyspark
import random

sc = pyspark.SparkContext(appName="Pi")
num_samples = 100000000

def inside(p):
    x, y = random.random(), random.random()
    return x*x + y*y < 1

count = sc.parallelize(range(0, num_samples)).filter(inside).count()

pi = 4 * count / num_samples
print(pi)

sc.stop()

3.14107536
```

Below the code cell, the prompt "In [9]:" is visible, followed by an empty input box.

Python REPL

```
$ pyspark
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
20/05/17 21:09:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to

  ____      _
 / ___|  _ \| | | |
 \___ \ |_) | | | |
  ___) |  __/| | | |
 |____|_|_||_| |_|
version 2.4.5

Using Python version 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019 21:01:17)
SparkSession available as 'spark'.
>>> |
```

```
$ pyspark
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
20/05/17 21:43:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to

  ____      _
 / ___|  _ \| | | |
 \___ \ |_) | | | |
  ___) |  __/| | | |
 |____|_|_||_| |_|
version 2.4.5

Using Python version 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019 00:11:34)
SparkSession available as 'spark'.
>>> |
```

R REPL

[illegible]

Resilient Distributed Datasets (RDDs)

- Τα RDDs μοιάζουν με τους πίνακες των Βάσεων Δεδομένων
 - Τα RDDs είναι immutable (ένα RDD μπορεί να τροποποιηθεί μέσω ενός μετασχηματισμού αλλά σε αυτή την περίπτωση επιστρέφεται ένα νέο RDD και το αρχικό RDD παραμένει το ίδιο)
-
- Τα RDDs υποστηρίζουν κατανεμημένη αποθήκευση δεδομένων στις μνήμες των μηχανημάτων ενός cluster έτσι ώστε να επιτυγχάνεται
 - **ανοχή σε σφάλματα:** καταγράφοντας το ιστορικό των μετασχηματισμών που εφαρμόζονται στα δεδομένα
 - **υψηλή απόδοση:** Παραλληλισμός επεξεργασίας στους κόμβους του cluster

RDDs: Transformations - Actions

Από τη στιγμή που έχει δημιουργηθεί ένα RDD, μπορούν να γίνουν δύο βασικοί τύποι λειτουργιών:

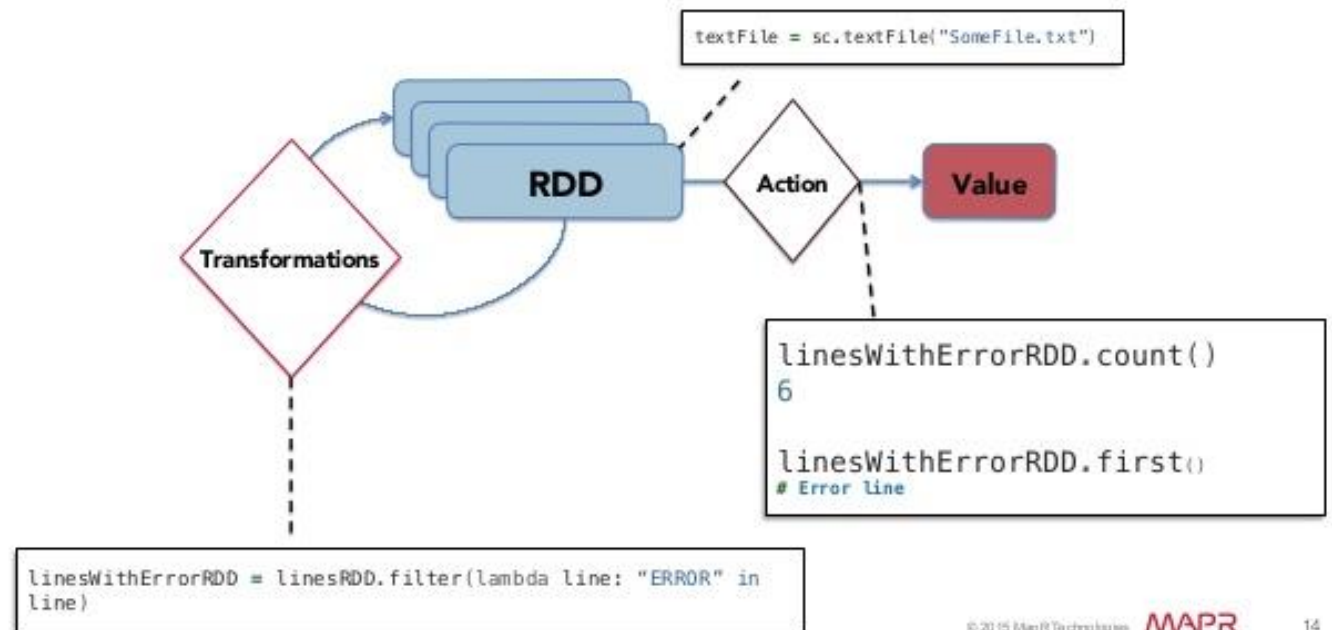
- **Μετασχηματισμοί (transformations)**
 - δημιουργούν ένα νέο RDD αλλάζοντας το αρχικό (π.χ. map, filter, flatMap, groupByKey, reduceByKey, aggregateByKey, pipe, coalesce)
 - οι μετασχηματισμοί δεν επιστρέφουν μια απλή τιμή αλλά ένα νέο RDD (lazy evaluation)
- **Ενέργειες (actions)**
 - υπολογίζουν μια ποσότητα αλλά δεν αλλάζουν τα δεδομένα (π.χ. reduce, collect, count, first, take, countByKey, foreach)
 - όταν καλείται μια ενέργεια σε ένα RDD, τότε εκτελούνται όλοι οι μετασχηματισμοί και επιστρέφεται το αποτέλεσμα

Directed Acyclic Graphs (DAGs)

- Το Spark επιτρέπει την ανάπτυξη σύνθετων εργασιών, που αποτελούνται από πολλά επιμέρους βήματα χρησιμοποιώντας το λεγόμενο DAG pattern
- Το Spark διατηρεί τα ενδιάμεσα αποτελέσματα στη μνήμη αντί να τα εγγράφει στο δίσκο (ιδιαίτερα χρήσιμο όταν χρειάζεται να πραγματοποιηθούν εργασίες στα ίδια δεδομένα πολλές φορές)



Working With RDDs



MLlib (scalable machine learning library)

- Το API του MLlib βασίζεται στα DataFrames
- Αλγόριθμοι μηχανικής μάθησης
 - Κατηγοριοποίηση
 - Παλινδρόμηση
 - Δένδρα απόφασης
 - Αλγόριθμοι συστάσεων
 - Συσταδοποίηση
 - ...

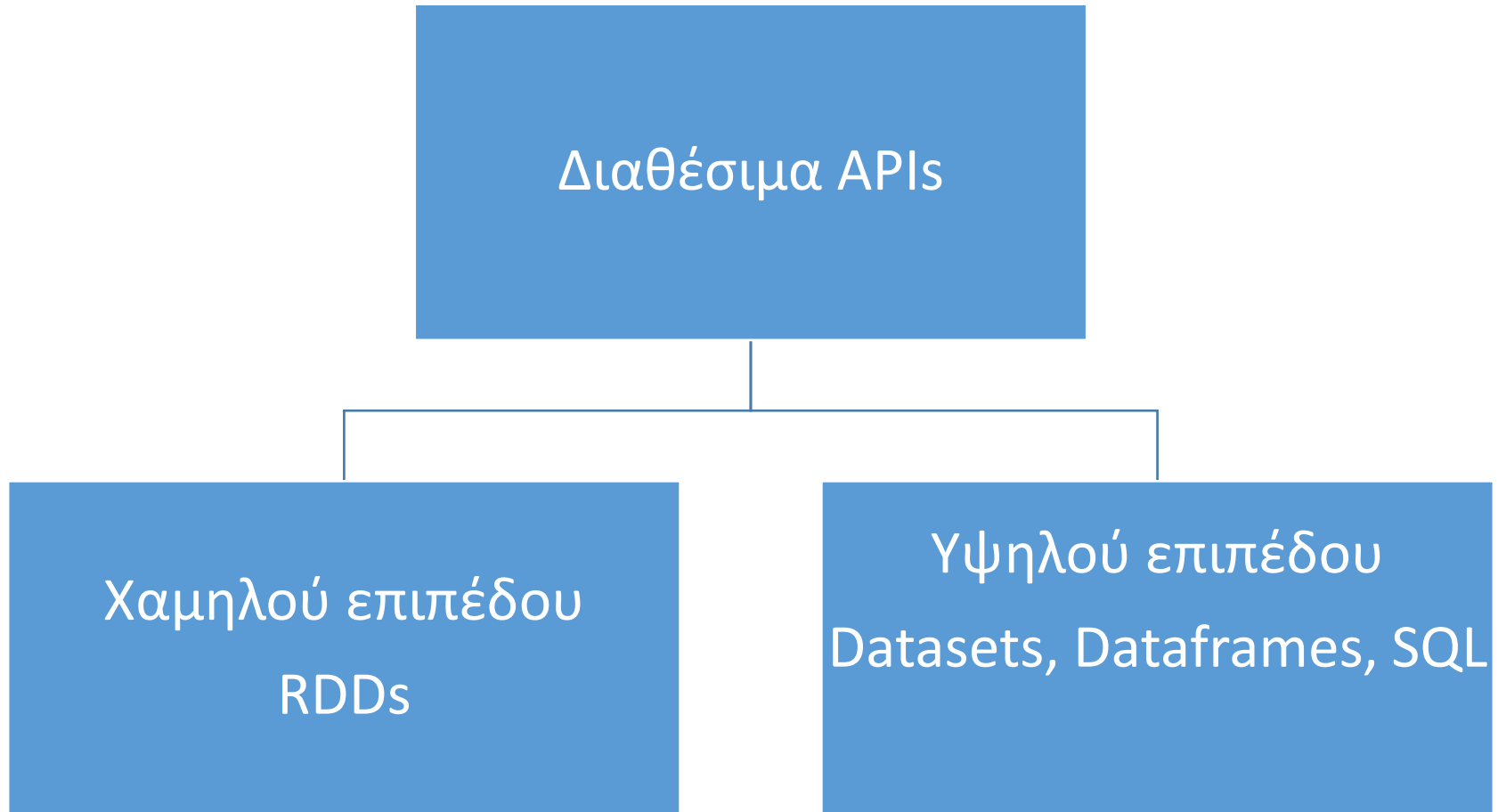
Streaming

- Τα streaming δεδομένα φθάνουν συνεχώς από διάφορες πηγές με μικρά μηνύματα
 - Υπάρχουν πολλές εφαρμογές της streaming τεχνολογίας (παρακολούθηση αισθητήρων, έλεγχος logs, παρακολούθηση χρηματοοικονομικών αγορών κ.α.)
-
- Λογισμικά ανάλυσης streams
 - Flink
 - Storm
 - Kafka
 - Spark
 - Samza
 - Kinesis
 - ...

Εξέλιξη του μοντέλου επεξεργασίας στο Apache Spark

- RDD (Resilient Distributed Datasets)
- Spark 1.3 ➔ DataFrame API (χρησιμοποιεί γλώσσα ερωτημάτων – query language – για να χειρίζεται τα δεδομένα ταχύτερα σε σχέση με τα RDD)
- Spark 1.6 ➔ DataSet API (δημιουργεί query plans για την εκτέλεση των ερωτημάτων, ταχύτερο σε σχέση με τα RDDs)
- Spark 2.0 ➔ Structured APIs (Datasets, DataFrames, SQL tables and views) – Schemas
- Spark 3.0 ➔ adaptive query execution; dynamic partition pruning; ANSI SQL compliance; significant improvements in pandas APIs; new UI for structured streaming; up to 40x speedups for calling R user-defined functions; accelerator-aware scheduler; and SQL reference documentation

Spark's APIs



Απόδοση του Apache Spark (GraySortMetric, CloudSortMetric, TPC-DS 30TB)

A

GraySort 2014	Hadoop MR Record	Spark Record
Data Size	102.5 TB	100 TB
Elapsed Time	72 mins	23 mins
# Nodes	2100	206
# Cores	50400 physical	6592 virtualized
Cluster disk throughput	3150 GB/s (est.)	618 GB/s
Sort Benchmark Daytona Rules	Yes	Yes
Network	dedicated data center, 10Gbps	virtualized (EC2) 10Gbps network
Sort rate	1.42 TB/min	4.27 TB/min
Sort rate/node	0.67 GB/min	20.7 GB/min

<https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html>

B

New CloudSort Benchmark

Cost to sort 100TB of data



<https://databricks.com/blog/2016/11/14/setting-new-world-record-apache-spark.html>

C

Spark 3.0 performed roughly 2x better than Spark 2.4 in total runtime for 30TB TPC-DS benchmark

<https://databricks.com/blog/2020/06/18/introducing-apache-spark-3-0-now-available-in-databricks-runtime-7-0.html>

Demo: παράδειγμα επεξεργασίας με RDDs στο Apache Spark

- Υπολογισμός πλήθους μοναδικών επισκεπτών ιστοσελίδας
- Εύρεση IP διευθύνσεων από τις οποίες συνδέθηκε ο κάθε μοναδικός χρήστης
- Χρήση αρχείων καταγραφής - weblogs (82.9MB)

Java magazine May/June 2016
Apache Spark 101: Getting up to speed on the popular big data engine

<http://www.oracle.com/technetwork/java/javamagazine>

id πελάτη

IP Διεύθυνση

```
3.94.78.5 - 69827 [15/Sep/2013:23:58:36 +0100]  
"GET /KBD0C-00033.html HTTP/1.0" 200 14417  
"http://www.loudacre.com"  
"Loudacre Mobile Browser iFruit 1"
```