



# Apache Hadoop

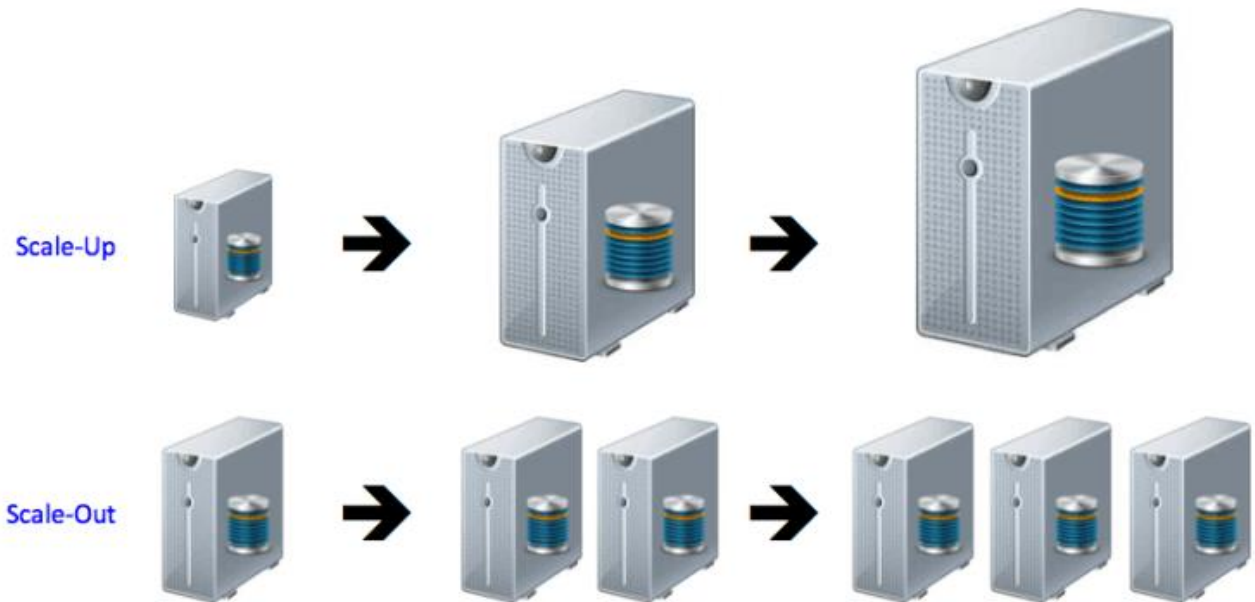
Χρήστος Γκόγκος

24/1/2022

[https://github.com/chgogos/big\\_data](https://github.com/chgogos/big_data)

# Το δίλημμα scale up ή scale out

- **Scale up:** Ένα μηχάνημα υψηλών προδιαγραφών και υψηλής αξιοπιστίας → εκθετική αύξηση κόστους
- **Scale out:** Μια συστάδα μηχανημάτων αποτελούμενη από απλούς υπολογιστές μέτριας αξιοπιστίας (commodity hardware) → γραμμική αύξηση κόστους



# Η έννοια της κλιμάκωσης στο λειτουργικό επίπεδο

Εφαρμογές που λειτουργούν εφόσον τα δεδομένα «χωράνε» στην κύρια μνήμη του υπολογιστή (απλά προγράμματα σε Python, R, Java, C, C++, ...)



Εφαρμογές που επιτρέπουν “out of core” επεξεργασία (βάσεις δεδομένων: Oracle, DB2, MySQL, ...)



Εφαρμογές που μπορούν να κάνουν χρήση μεγάλου αριθμού χαμηλού κόστους υπολογιστών (scale out)



# Η έννοια της κλιμάκωσης αλγοριθμικά

**Στο παρελθόν:** Αν η διάσταση του προβλήματος είναι  $n$  τότε ζητείται αλγόριθμος που να επιλύει το πρόβλημα σε πολυωνυμικό χρόνο (δλδ εκτελώντας το πολύ  $n^k$  λειτουργίες με  $k$  μια αριθμητική σταθερά)

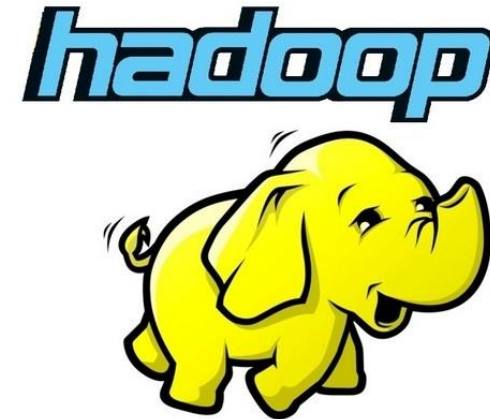
**Σήμερα:** Αν η διάσταση του προβλήματος είναι  $n$  τότε ζητείται αλγόριθμος που να επιλύει το πρόβλημα σε  $\frac{n^k}{c}$  χρόνο όπου  $c$  είναι ένα μεγάλο πλήθος υπολογιστών που μπορούν να διατεθούν (δλδ ζητούνται πολυωνυμικοί αλγόριθμοι που να μπορούν να παραλληλοποιηθούν)

**Στο μέλλον (σύντομα):** streaming δεδομένα (δεδομένα που έρχονται συνεχώς), είναι πρακτικό να γίνεται ένα μόνο πέρασμα από τα δεδομένα, αποδεκτή πολυπλοκότητα μπορεί να είναι το πολύ  $O(n \log n)$

# Apache Hadoop

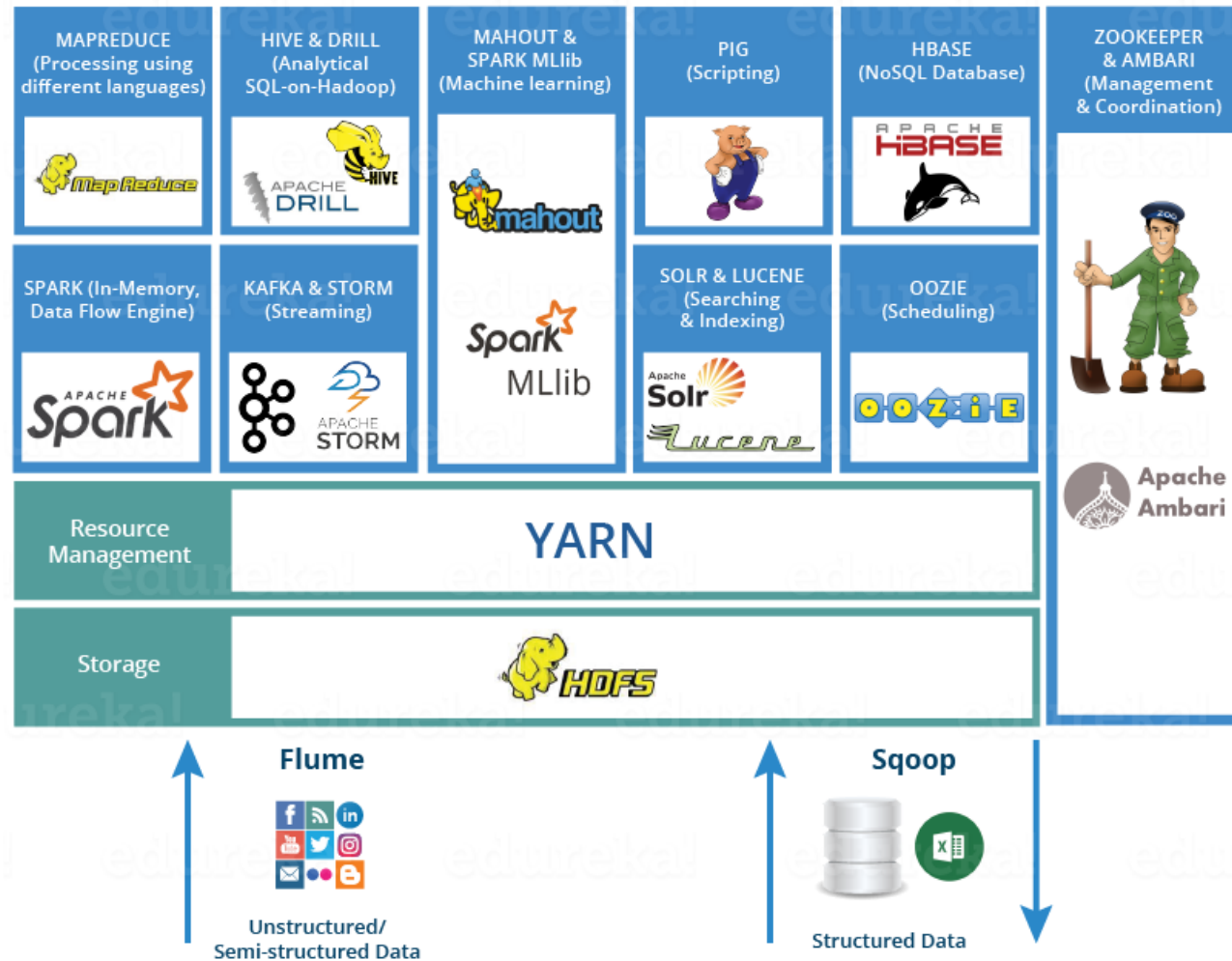
- Το Hadoop είναι ένα framework που επιτρέπει την αποθήκευση δεδομένων και την παράλληλη εκτέλεση εφαρμογών σε συστάδες απλών υπολογιστών (computer clusters)
- Βασικά υποσυστήματα του Hadoop είναι τα:
  - **HDFS**
  - **YARN**
  - **MapReduce**
- Είναι open source και έχει μεγάλη αποδοχή (Yahoo!, Twitter, Amazon, Facebook κ.α.)

Βασική ιδέα: Μετακίνηση των υπολογισμών στα δεδομένα



<http://hadoop.apache.org/>

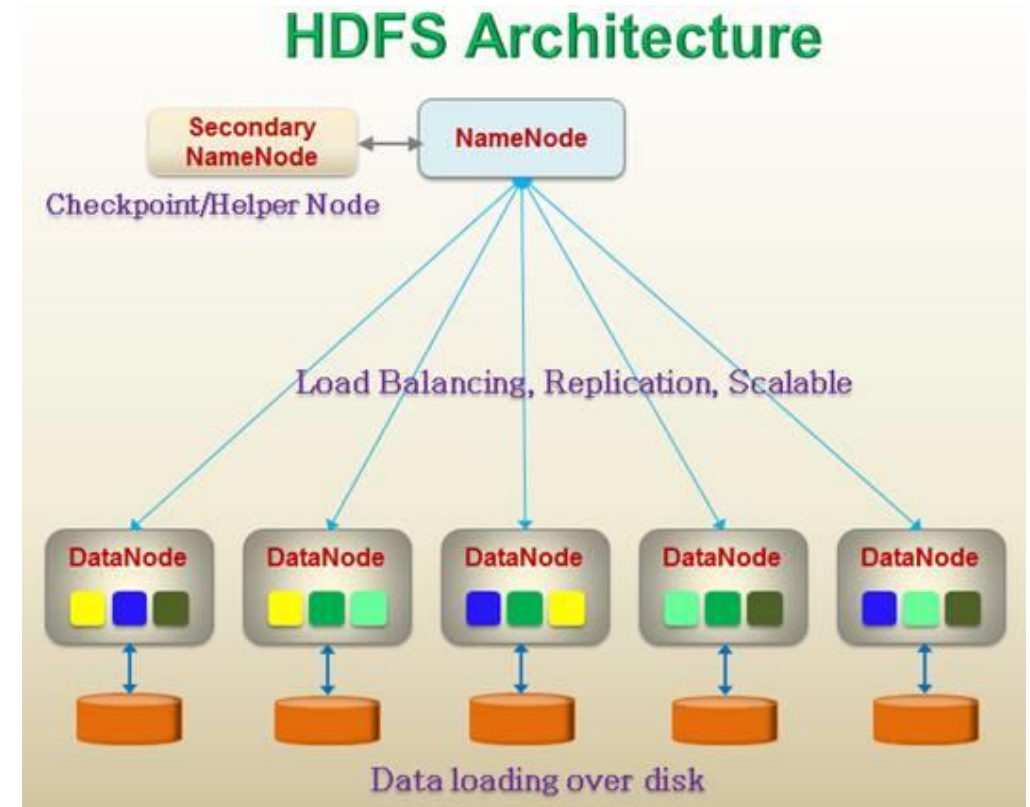
# Apache Hadoop Ecosystem



<https://www.edureka.co/blog/hadoop-ecosystem>

# HDFS (Hadoop Distributed File System)

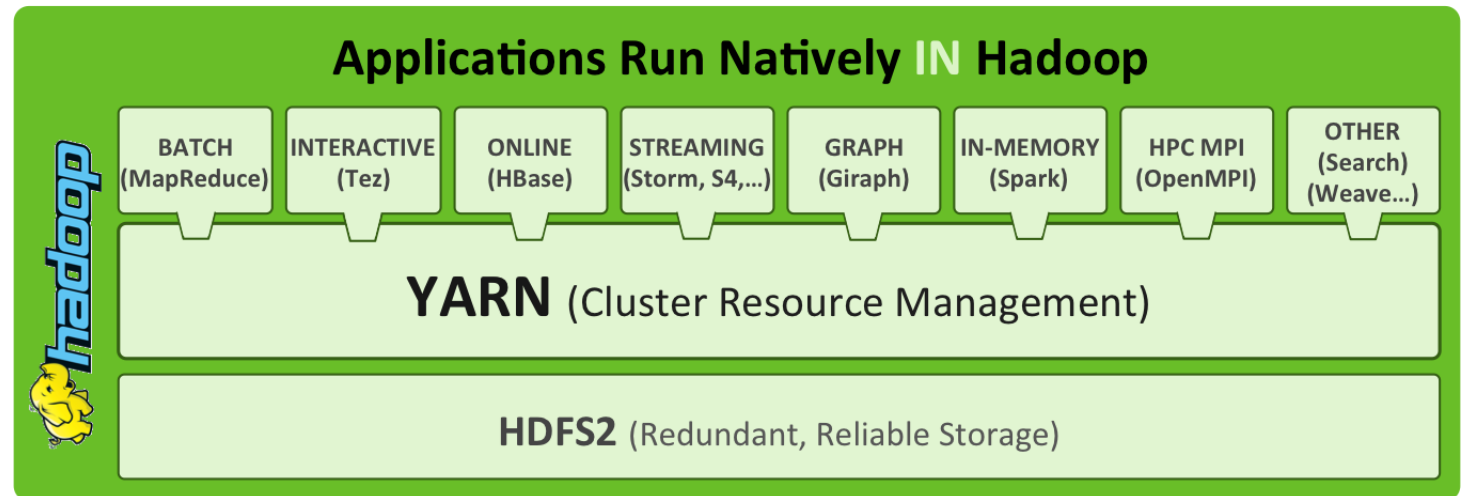
- Το HDFS είναι ένα **κατανεμημένο σύστημα αρχείων** που τρέχει πάνω από το σύστημα αρχείων ενός linux συστήματος
- Μπορεί να αποθηκεύσει μεγάλα αρχεία σε συστάδες κοινών υπολογιστών
- Τα αρχεία δεδομένων σπάνε σε μπλοκς (128 MB)
- Κάθε μπλοκ διατηρείται σε 3 (default) αντίγραφα σε διάφορους υπολογιστές



<http://www.developer.com/db/getting-familiarized-with-the-hadoop-distribution-file-system.html>

# Hadoop YARN (Yet Another Resource Negotiator)

- Το YARN λειτουργεί ως ενδιάμεσο επίπεδο ανάμεσα στις εργασίες που πρόκειται να εκτελεστούν και στο HDFS
- Φροντίζει ώστε να γίνεται καλή χρήση της υπολογιστικής ισχύος της υποδομής
- JobTracker (master)
- TaskTrackers (slaves) – ένα ανά datanode





# MapReduce

- Το MapReduce (Google 2004) είναι ένα υπολογιστικό υπόδειγμα (computational paradigm/engine) που χρησιμοποιείται ευρύτατα για αποδοτική καταναεμημένη επεξεργασία πάνω σε μεγάλα σύνολα δεδομένων
- Εκτελείται σε clusters υπολογιστών και επωφελείται από την ύπαρξη πολλών κόμβων στους οποίους μπορεί να ανατεθεί εργασία
- Ο προγραμματιστής χρειάζεται να γράφει δύο συναρτήσεις για την αντιμετώπιση κάθε προβλήματος με MapReduce: τη συνάρτηση map και τη συνάρτηση reduce
- Το Hadoop MR αποτελεί μια open source υλοποίηση του MapReduce που περιλαμβάνεται στο Apache Hadoop

Δεκέμβριος 2004

## MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

*Google, Inc.*

### Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

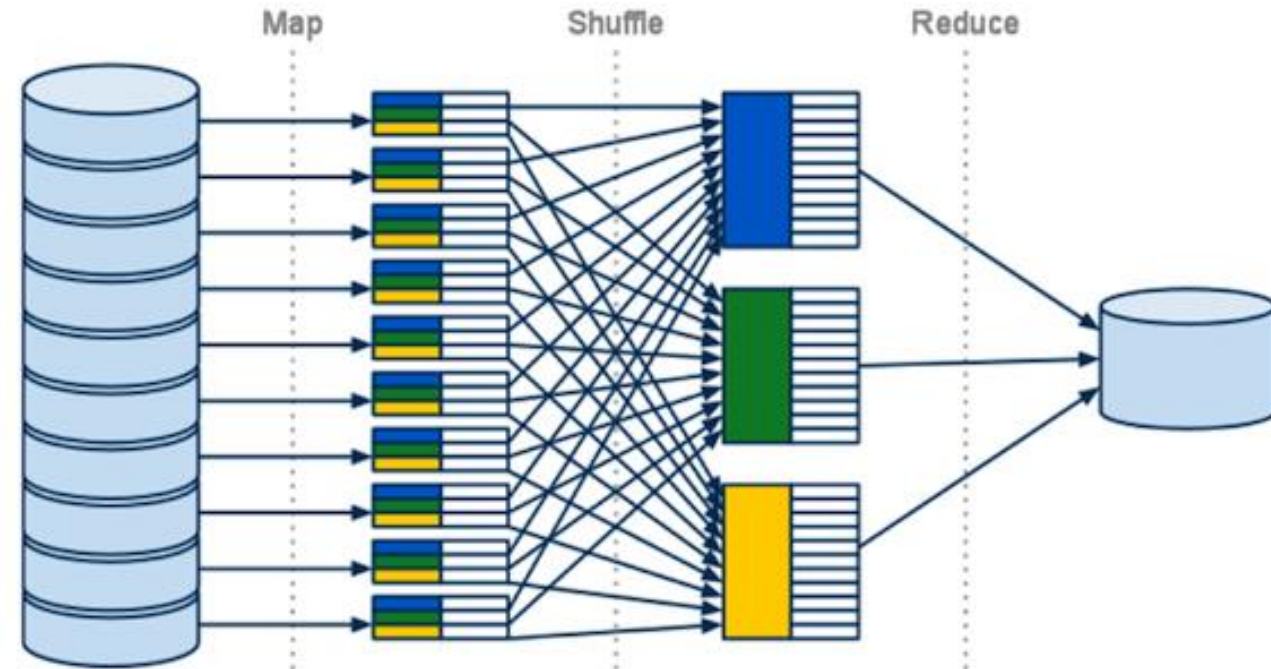
given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

As a reaction to this complexity, we designed a new

<http://research.google.com/archive/mapreduce-osdi04.pdf>

# Οι συναρτήσεις map και reduce του MapReduce

- Τα προβλήματα «σπάνε» σε δύο φάσεις
  - **Map:** τα δεδομένα του προβλήματος διαχωρίζονται σε μη επικαλυπτόμενα τμήματα της μορφής <key, value> και ανατίθενται σε διεργασίες που παράγουν αποτελέσματα επίσης της μορφής <key, value>
  - **Shuffle (and sort):** γίνεται από το Hadoop MR – ταξινομεί και ομαδοποιεί τα δεδομένα της map φάσης
  - **Reduce:** τα αποτελέσματα της Map φάσης τροφοδοτούνται σε διεργασίες που τα συνοψίζουν σε μικρότερο αριθμό εγγραφών



<https://developers.google.com/appengine/docs/python/dataprocessing/>

# Παράδειγμα Map-Reduce: Καταμέτρηση λέξεων

- Παράδειγμα προβλήματος: Υπολογισμός της συχνότητας εμφάνισης λέξεων σε ένα σύνολο πολλών κειμένων
- Ο προγραμματιστής πρέπει να γράψει μια **map συνάρτηση** και μια **reduce συνάρτηση**
- Το σύστημα ομαδοποιεί όλα τα ζεύγη με το ίδιο ενδιαμέσο κλειδί και περνά το σύνολο των ζευγών στη συνάρτηση reduce

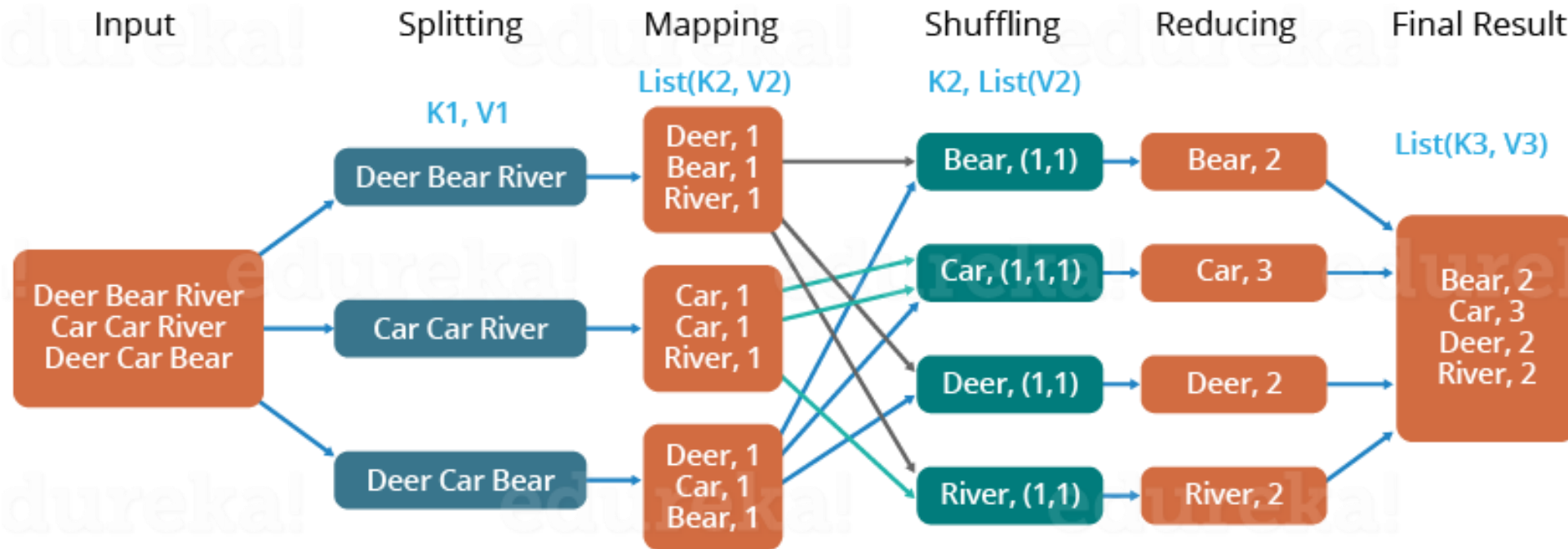
```
// key: όνομα του εγγράφου  
// value: περιεχόμενο του εγγράφου  
map(String key, String value)  
for each w in value  
    emitIntermediate(w,1)
```

```
// key: μια λέξη  
// values: μια λίστα από μετρήσεις  
reduce(key, values)  
c = 0  
for each v in values  
    c += v  
emit(c)
```

# Παράδειγμα καταμέτρησης λέξεων (σε 1 κείμενο)

## The Overall MapReduce Word Count Process

edureka!



<https://www.edureka.co/blog/mapreduce-tutorial>

# Βασικά βήματα του MapReduce

1. Φόρτωση ενός μεγάλου συνόλου εγγραφών σε ένα σύνολο μηχανημάτων με τη μορφή (key, value)
  2. Εξαγωγή κάποιας χρήσιμης πληροφορίας από κάθε εγγραφή (map) και εκπομπή ενδιάμεσου αποτελέσματος με τη μορφή (intermediate\_key, value)
  3. Ανακάτεμα (shuffle) των ενδιάμεσων αποτελεσμάτων ανάμεσα στα μηχανήματα έτσι ώστε το ενδιάμεσα αποτελέσματα για το ίδιο κλειδί να είναι μαζί
  4. Υπολογισμός συγκεντρωτικών τιμών για τα ενδιάμεσα αποτελέσματα (reduce)
  5. Αποθήκευση τελικού αποτελέσματος
- **Βήμα Map**
    - Τα δεδομένα εισόδου χωρίζονται σε τμήματα
    - Οι worker nodes επεξεργάζονται παράλληλα τα δεδομένα που έχουν αναλάβει
    - Κάθε worker node αποθηκεύει το αποτέλεσμα στο τοπικό του σύστημα αρχείων από όπου ο reducer μπορεί να το προσπελάσει
  - **Βήμα Reduce**
    - Πολλαπλά reduce tasks αναλαμβάνουν να υπολογίσουν παράλληλα συγκεντρωτικά αποτελέσματα

# Διαχωρισμός εργασιών: Τι κάνει ο προγραμματιστής και τι το Hadoop

## **Προγραμματιστής**

- Γράφει τη συνάρτηση map
- Γράφει τη συνάρτηση reduce
- Γράφει το driver πρόγραμμα

## **Framework Hadoop**

- Ανοχή σε σφάλματα
- Ανάθεση workers σε map και reduce
- Μετακίνηση επεξεργασίας στα δεδομένα
- Shuffle & sort ανακάτεμα και ταξινόμηση των δεδομένων

# Τρόποι προγραμματισμού στο Hadoop

- **Java:** πρόσβαση στις πλήρεις δυνατότητες του MapReduce αλλά δύσκαμπτα και μεγάλα προγράμματα
- **Hadoop streaming:** δυνατότητα χρήσης άλλων γλωσσών (π.χ. Python, Ruby κ.α.)
- **Cascading:** βιβλιοθήκη σε java που επιτρέπει την περιγραφή data flows που μετασχηματίζονται σε διεργασίες map reduce
- **Pig:** γλώσσα data flow
- **Scalding:** γλώσσα βασισμένη στη Scala που οδηγεί στη συγγραφή σύντομων προγραμμάτων
- **Hive:** γλώσσα παρόμοια με την SQL

## Hadoop Streaming



# Η θέση του Hadoop στην αγορά

- Hadoop Market by Product (Software, Hardware and Services) and by Applications - Global Industry Analysis and Forecast 2018 – 2024

<https://www.marketresearchengine.com/reportdetails/hadoop-market-share>

- Η αγορά του Hadoop αναμένεται να ξεπεράσει τα 50 δισεκατομμύρια \$ μέχρι το 2024



## ΘΕΤΙΚΑ

- Γρήγορη και οικονομική επεξεργασία δεδομένων
- Χειρισμός τεράστιων ποσοτήτων αδόμητων δεδομένων
- Αυξημένη ζήτηση για data analytics

## ΑΡΝΗΤΙΚΑ

- Ασφάλεια – ιδιωτικότητα
- Χαμηλή διαθεσιμότητα εξειδικευμένου προσωπικού



# Major Hadoop vendors

- Cloudera
  - Hortonworks
- MapR

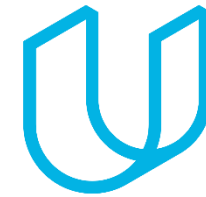
<https://www.zdnet.com/article/the-new-cloudera-hortonworks-hadoop-100-open-source-50-boring/>



- Hadoop sandbox Virtual Machines (VMs)
  - Κάθε VM είναι «φορτωμένο» με ένα μεγάλο αριθμό από εργαλεία
  - Απαιτούν μεγάλη ποσότητα μνήμης
    - cloudera CDH 5.8 (4+ GB RAM – 2 cores)
    - Hortonworks HDP 2.6.3 (8+ GB RAM – 4 cores)

# Demo: παράδειγμα επεξεργασίας με το Hadoop Map Reduce

- Υπολογισμός των συνολικών πωλήσεων ανά πόλη από ένα αρχείο με δεδομένα πωλήσεων purchases.txt (202MB)
- Εκτέλεση κώδικα στο Virtual Machine CDH 4.1.1 της Cloudera
- Συγγραφή κώδικα σε python (mapper.py, reducer.py) που με το **Hadoop streaming** μετατρέπεται σε jar αρχεία java



UDACITY

Intro to Hadoop and MapReduce by Cloudera

<https://www.udacity.com/course/intro-to-hadoop-and-mapreduce--ud617>