



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.



Τμήμα Μηχανικών  
Πληροφορικής ΑΤΕΙΘ



Υλοποίηση αλγορίθμων  
ταξινόμησης μεγάλου  
όγκου δεδομένων με το  
μοντέλο MapReduce

Πτυχιακή Εργασία  
Φοιτητής: Γαβριηλίδης  
Αχιλλέας

Επιβλέπων Καθηγητής:  
Κωνσταντίνος Διαμαντάρας

# Big Data

- Big Data είναι μία συλλογή από dataset τόσο μεγάλων και πολύπλοκων, που γίνεται δύσκολο να επεξεργαστούν χρησιμοποιώντας παραδοσιακά εργαλεία διαχείρισης βάσεων δεδομένων.
- Τρεις τρόποι επεξήγησης των Big Data είναι μέσω 3 βασικών ιδιοτήτων ( 3 V's):
  - Όγκος δεδομένων (Volume)
  - Ταχύτητα παραγωγής δεδομένων (Velocity)
  - Ποικιλία των δεδομένων (Variety)

# Big Data - Volume

- Ο όγκος των δεδομένων, έχει αυξηθεί δραματικά τα τελευταία χρόνια

1.3 Billion RFID tags in 2005  
**30 Billion** RFID today



Capital market  
data volumes grew  
**1,750%**, 2003-06



World Data Centre for Climate  
▪ **220 Terabytes** of Web data  
▪ **9 Petabytes** of additional data



**2 Billion** Internet  
users by 2011



**4.6 Billion**  
Mobile Phones  
World Wide



Twitter process  
**7 terabytes** of  
data every day

**facebook**

Facebook processes  
**10 terabytes** of  
data every day

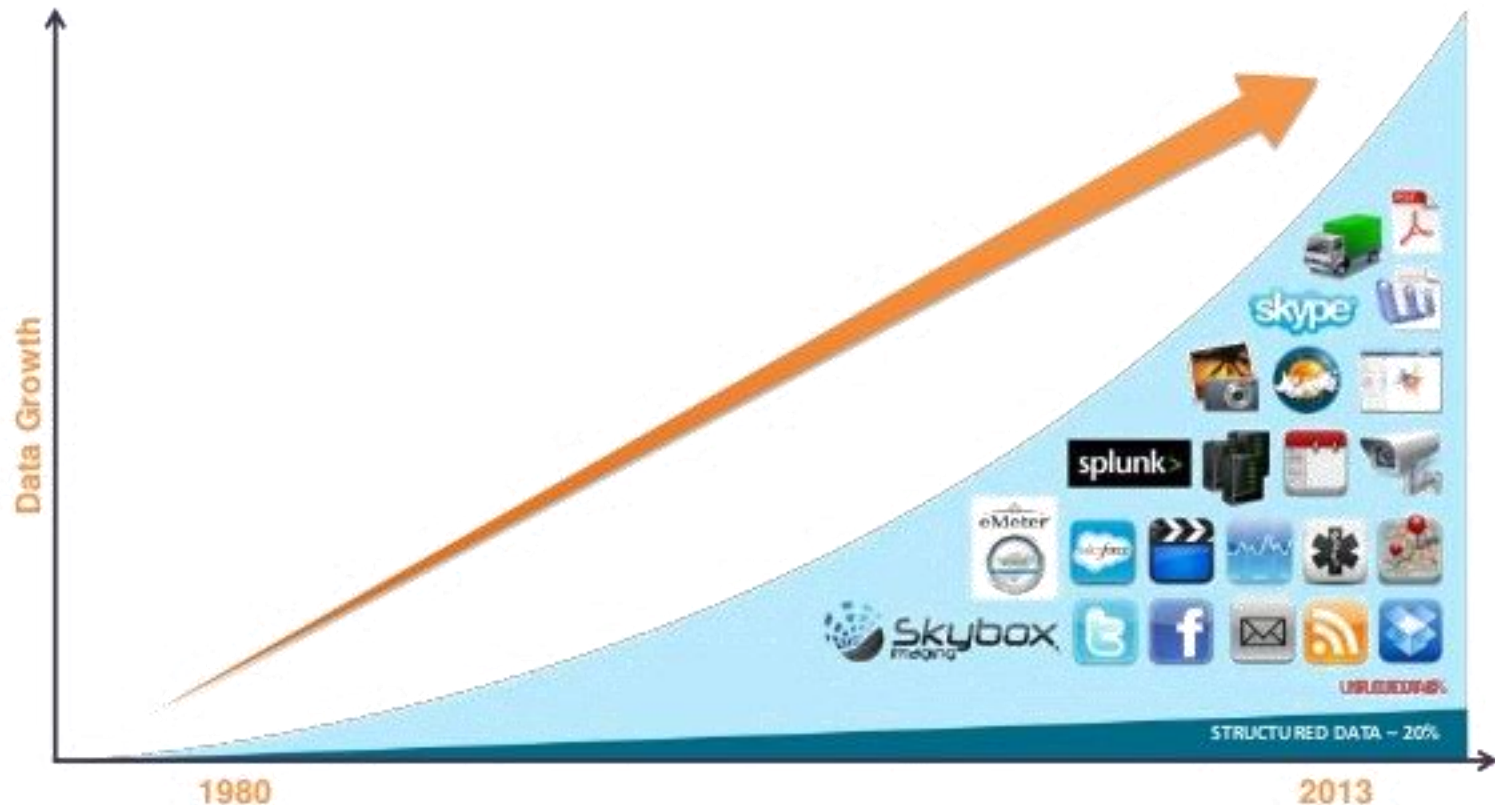


# Big Data – Volume (2)

- 2,8 ZetaByte το 2012
- 85% από νέους τύπους δεδομένων
- 15x Machine Data μέχρι το 2020
- 40 ZetaByte μέχρι το 2020

# Big Data - Velocity

- Η παραγωγή νέων δεδομένων έχει αυξηθεί εκθετικά.

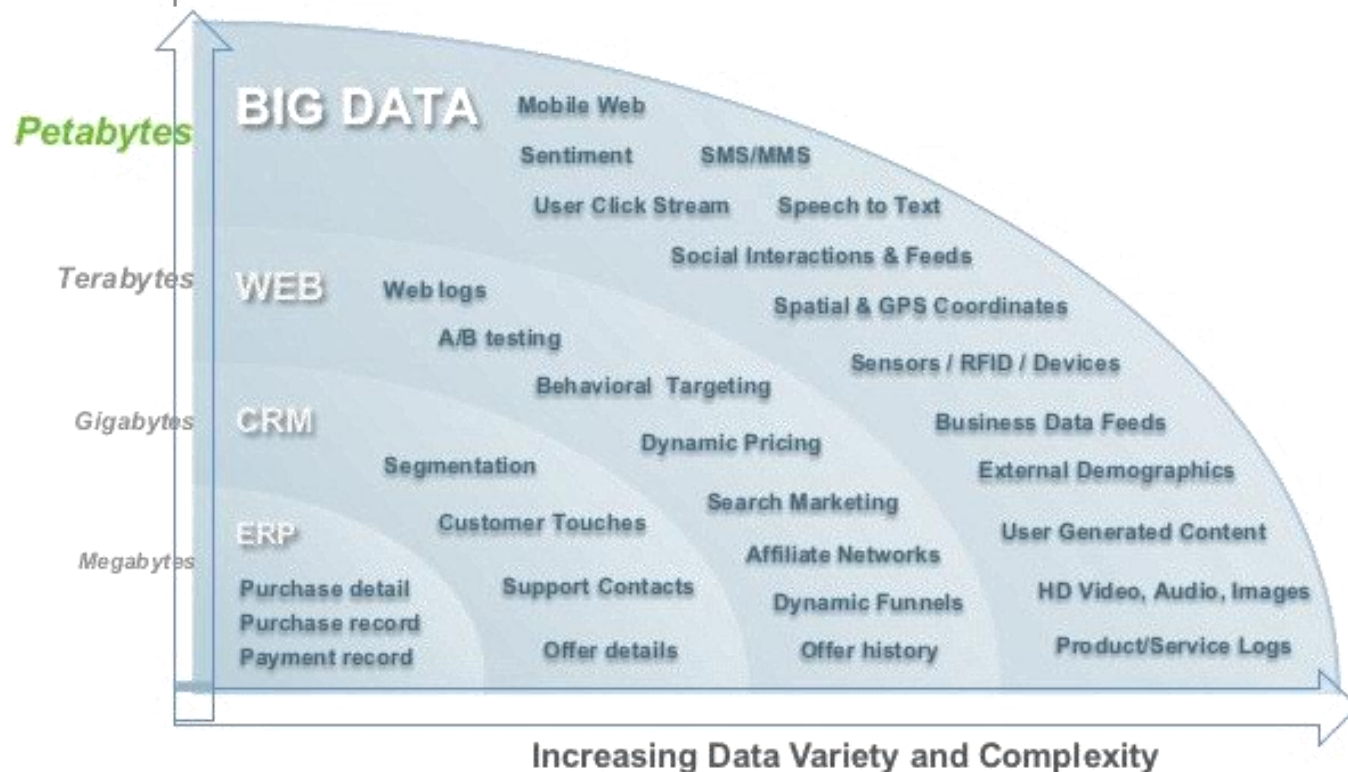


# Big Data – Velocity (2)

- Πλέον με την ευρεία χρήση του Διαδικτύου και των νέων τεχνολογιών, παράγονται νέα δεδομένα κάθε δευτερόλεπτο.
- Κάθε 60 δευτερόλεπτα: (πηγή: [archive.dzone.com](http://archive.dzone.com))
  - 694,445 ερωτήματα προς την Google
  - 6,600 φωτογραφίες ανεβαίνουν στο Flickr
  - 600+ βίντεο ανεβαίνουν στο YouTube
  - 70 νέα domain καταχωρούνται
  - 168,000,000 email στέλνονται
  - 320 λογαριασμοί Twitter και 98,000 Tweets δημιουργούνται
  - 370,000+ λεπτά κλήσεων Skype
  - 100 λογαριασμοί στο LinkedIn
  - 695,000 status updates, 79,634 wall posts και 510,040 σχόλια στο Facebook (παράγει μισό Petabyte κάθε μέρα)

# Big Data - Variety

- Πλέον υπάρχει μεγάλη ποικιλία δεδομένων, τα οποία χρειάζονται επεξεργασία και δεν είναι κατάλληλα για τις κλασικές βάσεις δεδομένων (π.χ. δεδομένα αισθητήρων, ελεύθερο κείμενο, εικόνες, βίντεο, 3D μοντέλα κ.α.). Έτσι υπάρχει η ανάγκη για επεξεργασία διαφόρων τύπων δεδομένων.

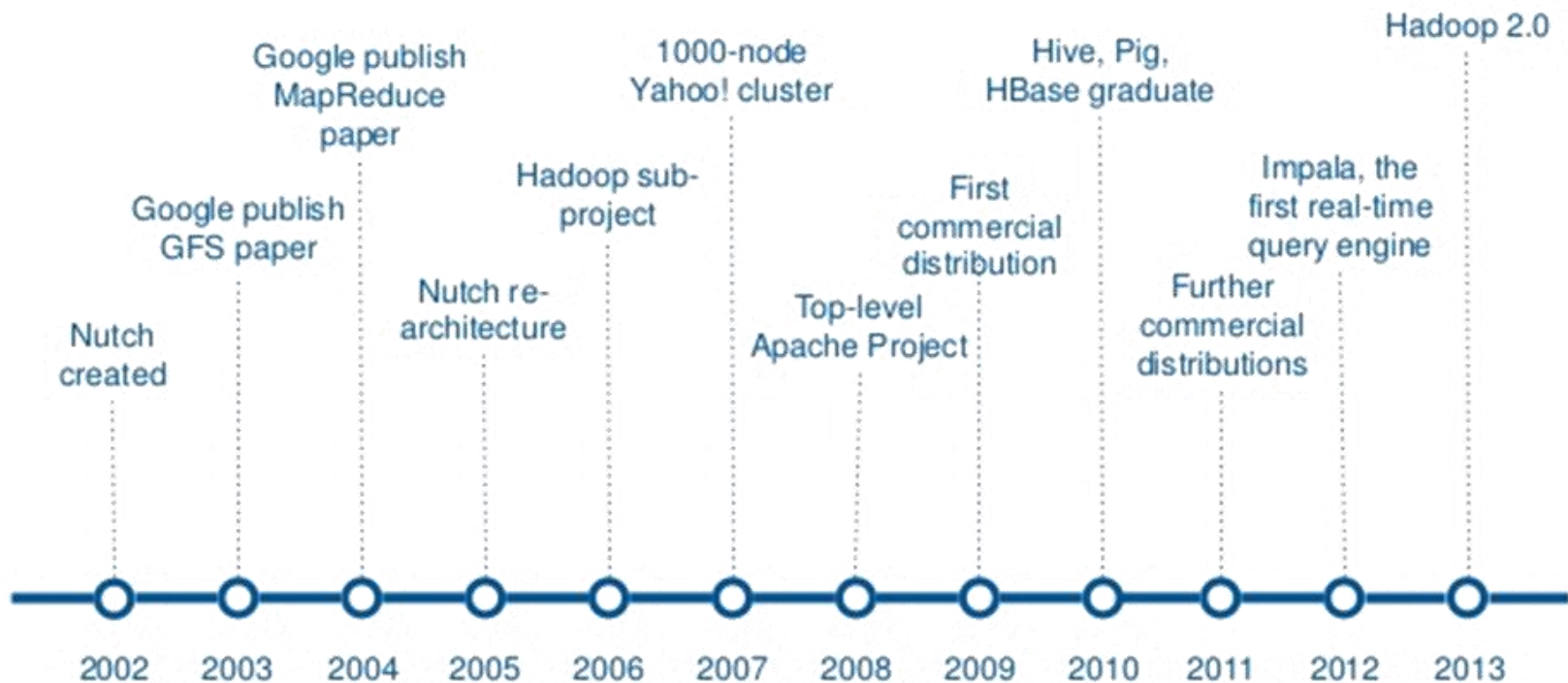


# Ιστορία του Apache Hadoop

- Ένα σύστημα κατανομημένης αποθήκευσης και επεξεργασίας.
  - Open Source.
  - Επεκτάσιμο.
  - Ανθεκτικό σε σφάλματα.
  - Αποθηκεύει από TeraByte έως και PetaByte.
  - Επεκτείνεται σε χιλιάδες κόμβους.
  - Χρησιμοποιεί χαμηλού κόστους υλικό (commodity hardware).
  - Αυτόματη ανάκτηση από σφάλματα.
  - Απλό μοντέλο προγραμματισμού.
- Ξεκίνησε από το project αναζήτησης Nutch και από τα paper της Google (GFS - 2003, MapReduce – 2004, BigTable - 2006).



# Ιστορία του Apache Hadoop (2)

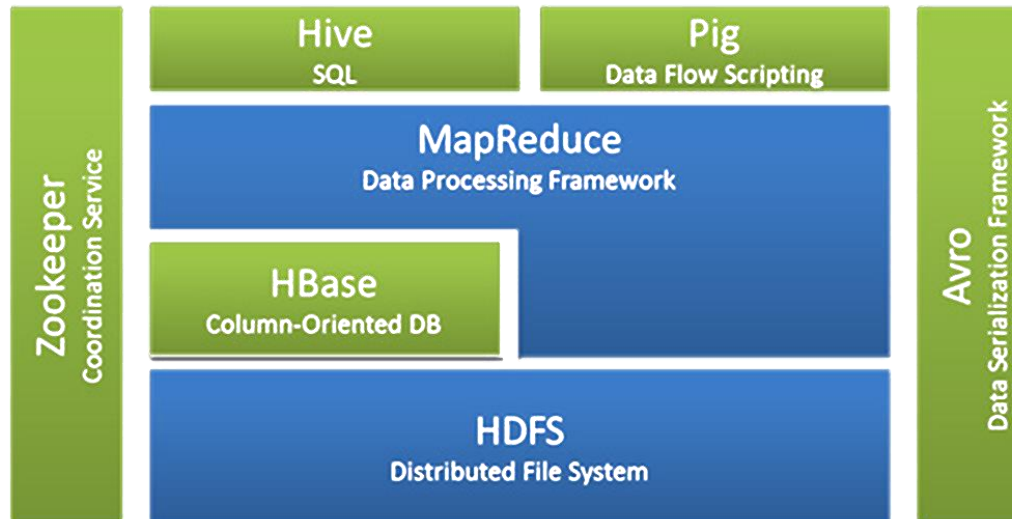


# Βασικές ιδέες - Apache Hadoop

- Οι εφαρμογές γράφονται σε κώδικα υψηλού επιπέδου
  - Οι προγραμματιστές δεν χρειάζεται να ανησυχούν για την επικοινωνία μεταξύ των κόμβων κλπ.
- Όσο το δυνατόν λιγότερη επικοινωνία μεταξύ των κόμβων.
- Μετακίνηση του υπολογισμού στα δεδομένα, όχι το αντίθετο.
  - Η επεξεργασία γίνεται τοπικά σε κάθε μηχάνημα στα δεδομένα που διαθέτει.
- Τα σφάλματα είναι αποδεκτά και αντιμετωπίζονται.
  - Τα δεδομένα αντιγράφονται πολλαπλές φορές σε διαφορετικά μηχανήματα.

# Apache Hadoop

- Δύο βασικά μέρη του Apache Hadoop είναι:
  - Το HDFS (Hadoop Distributed File System)
    - Ένα κατανεμημένο σύστημα αρχείων που παρέχει υψηλής απόδοσης πρόσβαση στα δεδομένα.
  - Το Hadoop MapReduce
    - Ένα framework (software) για κατανεμημένη επεξεργασία μεγάλων dataset σε cluster.

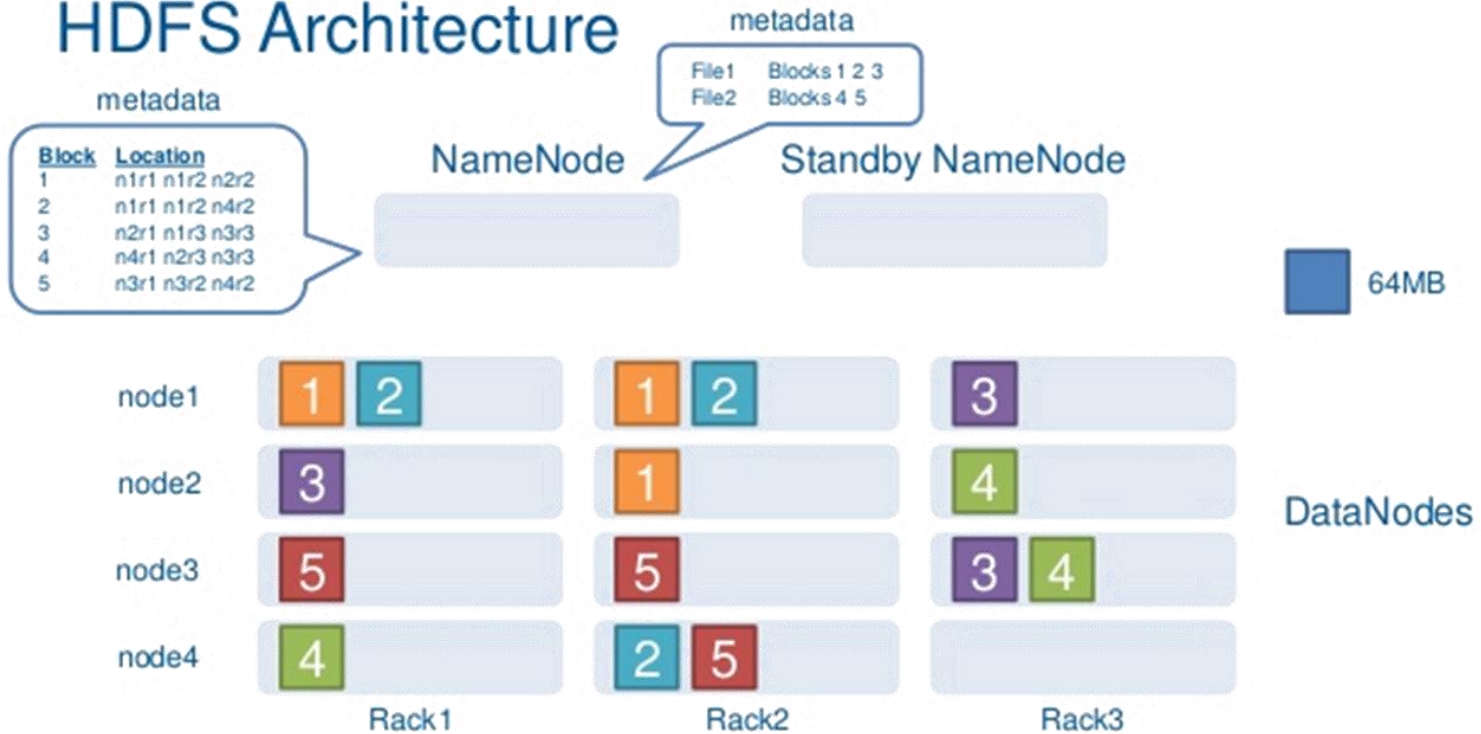


# Apache Hadoop - HDFS

- Κατανεμημένο, επεκτάσιμο, φορητό σύστημα αρχείων γραμμένο σε Java για το Hadoop.
- Λειτουργεί στο υπάρχον σύστημα αρχείων.
- Βασισμένο στο Google File System.
- Μπορεί να λειτουργήσει και ανεξάρτητα από το Hadoop.
- Υψηλό throughput.
- Υποστηρίζει πολύ μεγάλα αρχεία.
- Διαχωρίζει τα αρχεία σε μικρότερα block (64MB, πλέον 128MB ή και 256MB).
- Αξιοπιστία και ανοχή σε λάθη χάρη στην αντιγραφή των δεδομένων σε πολλούς κόμβους.
- Επίγνωση τοποθεσίας (rack awareness).
- Αποτελείται από:
  - NameNode (Master) για τα metadata.
  - Secondary NameNode (Master).
  - DataNode (Slave) τα πραγματικά δεδομένα.

# Apache Hadoop – HDFS (2)

## HDFS Architecture

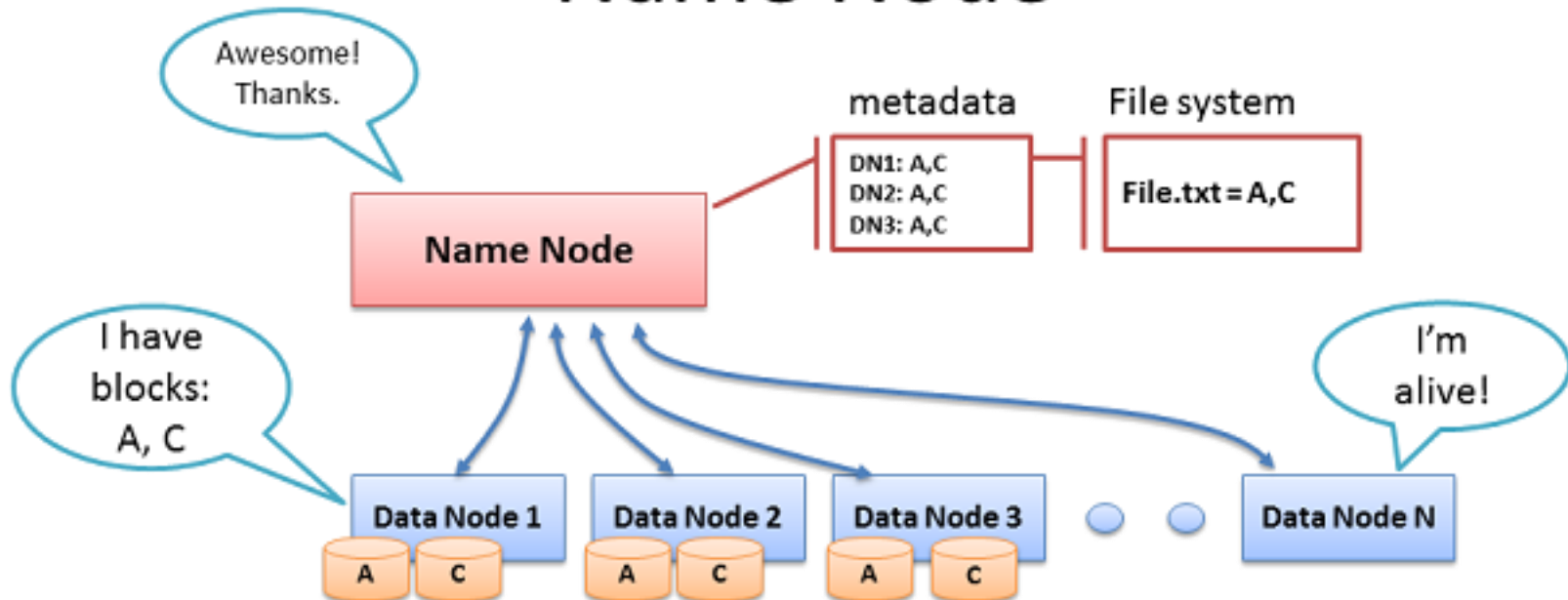


# Apache Hadoop – HDFS

## NameNode

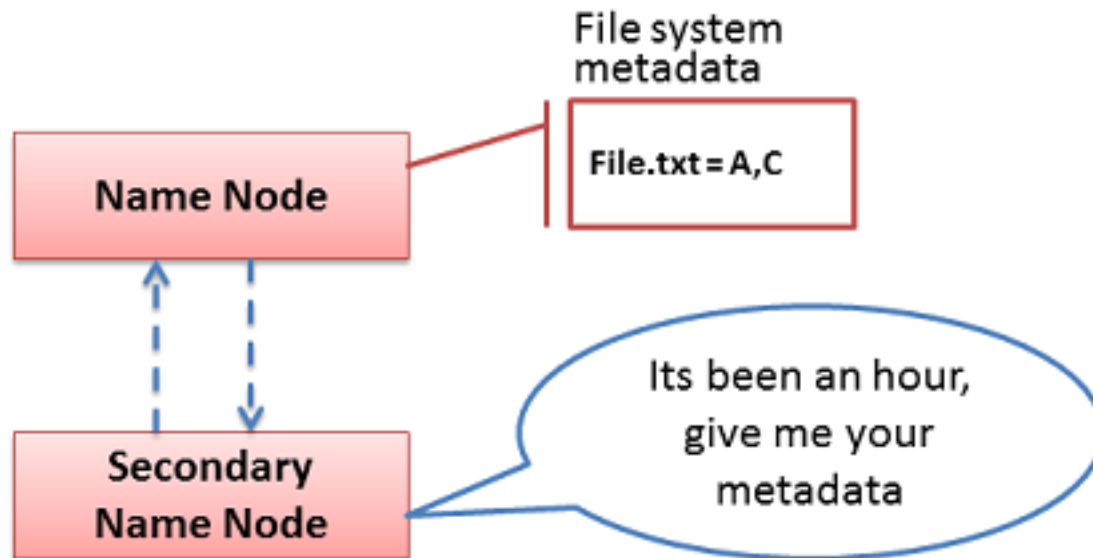
- Κάθε DataNode στέλνει ένα Heartbeat.
- Κάθε 10<sup>ο</sup> Heartbeat, είναι μια αναφορά για τα block που έχει.
- Ο NameNode φτιάχνει τα metadata από της αναφορές των block.

## Name Node



# Apache Hadoop – HDFS Secondary NameNode

## Secondary Name Node



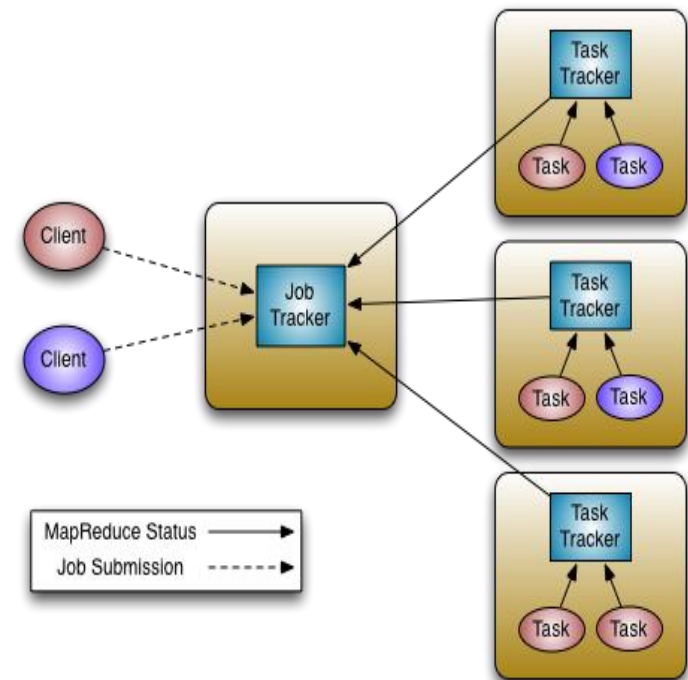
# Apache Hadoop - MapReduce

- Είναι ένα μοντέλο προγραμματισμού, εμπνευσμένο από τις συναρτήσεις `map()` και `reduce()` του συναρτησιακού προγραμματισμού.
- Επεξεργάζεται ζεύγη της μορφής `<key, value>`.
- Ιδανικό για επεξεργασία πολύ μεγάλων dataset.
- Αυτόματη παραλληλοποίηση και κατανομή της εργασίας.
- Αυτόματη ανάκτηση από σφάλματα όπως και το HDFS (ανάθεση εργασίας σε άλλο κόμβο).
- Σχετικά εύκολο στον προγραμματισμό (συνήθως με Java ή σε άλλες γλώσσες μέσω του Hadoop Streaming API).
- Εργαλεία για παρακολούθηση της κατάστασης της εργασίας.
- Χωρίζει την επεξεργασία σε 2 φάσεις:
  - Map φάση.
  - Reduce φάση.



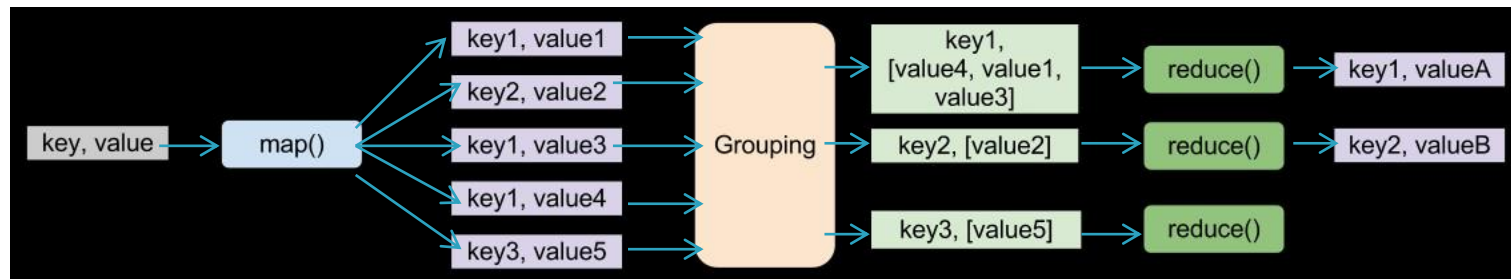
# Apache Hadoop – MapReduce (2)

- JobTracker
  - Διαχείριση πόρων του cluster.
    - Διαθέσιμοι TaskTracker
    - Map και Reduce slot.
  - Χρονοδρομολογεί της εργασίες.
    - Χρονοδρομολογεί όλα τα task που ανήκουν σε μία εργασία.
    - Παρακολουθεί την εκτέλεση.
    - Επανεκκινεί αποτυχημένες εργασίες.
    - Υπολογίζει τα σύνολα των counter των εργασιών.
- TaskTracker
  - Ένας σε κάθε κόμβο.
  - Εκτελεί τις map και reduce εργασίες.
  - Ειδοποιεί τον JobTracker
    - Για τον αριθμό των slot.
    - Για την κατάσταση της εκτέλεσης.
    - Στέλνει heartbeat για να ειδοποιήσει για την κατάσταση του.



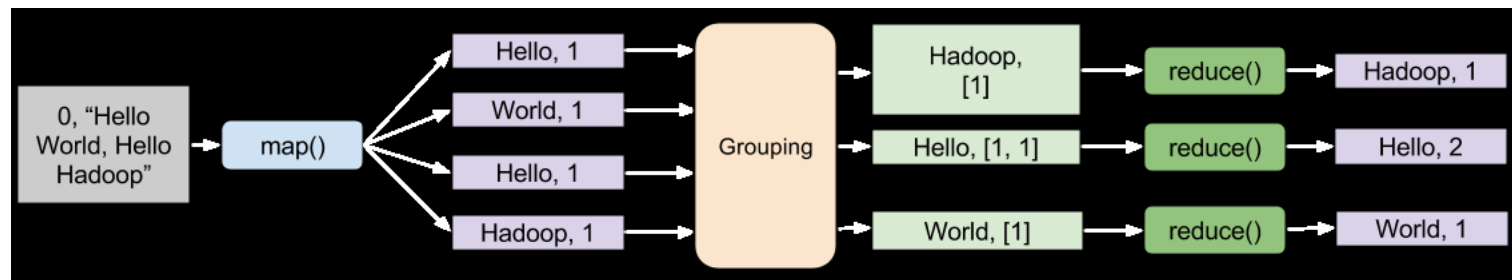
# Apache Hadoop – MapReduce (3)

- Map φάση:
  - Ο Master κόμβος παίρνει ένα πρόβλημα και το χωρίζει σε υπό-προβλήματα και τα κατανέμει στους worker κόμβους.
  - Οι worker κόμβοι μπορούν να το επαναλάβουν.
  - Το αποτέλεσμα όλων των map() ταξινομείται με βάση το κλειδί.
  - Μετά την εκτέλεση στέλνουν το αποτέλεσμα στον Master κόμβο.

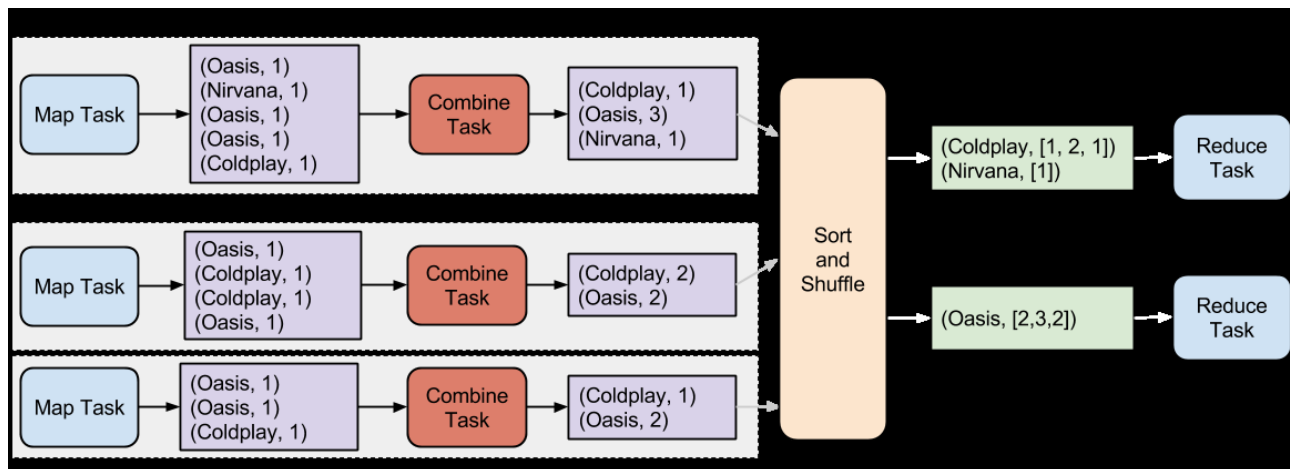
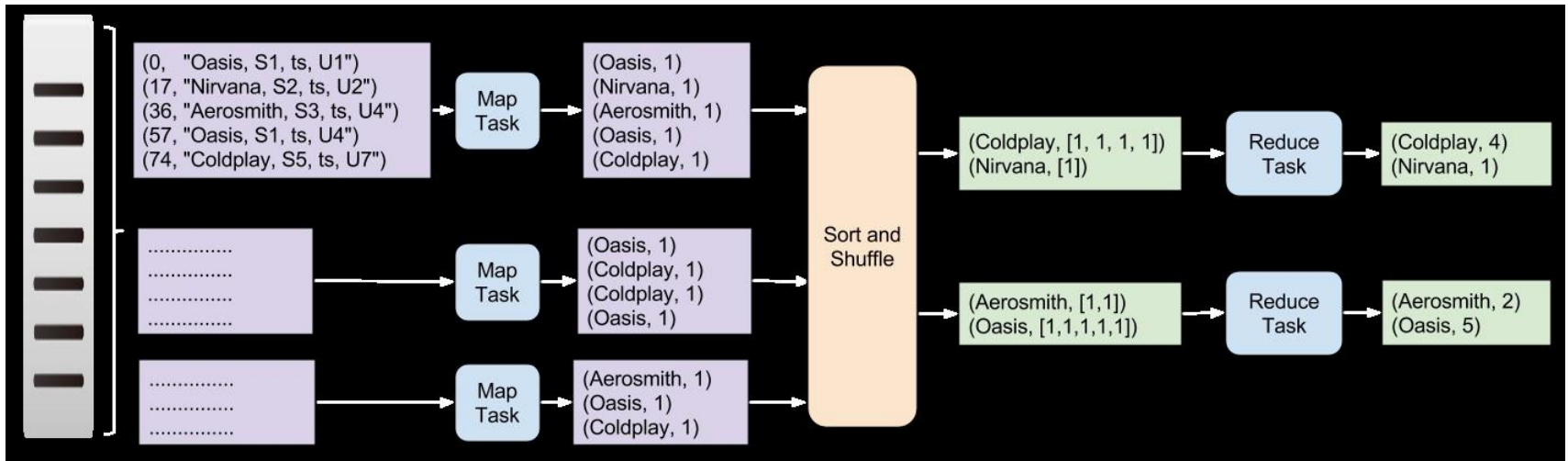


# Apache Hadoop – MapReduce (4)

- Reduce φάση:
  - Η `reduce()` καλείται για κάθε κλειδί.
  - Ο Master κόμβος λαμβάνει το αποτέλεσμα για κάθε υπό-πρόβλημα και τα συνδυάζει με ένα προκαθορισμένο τρόπο για να πάρει το αποτέλεσμα για το αρχικό πρόβλημα.

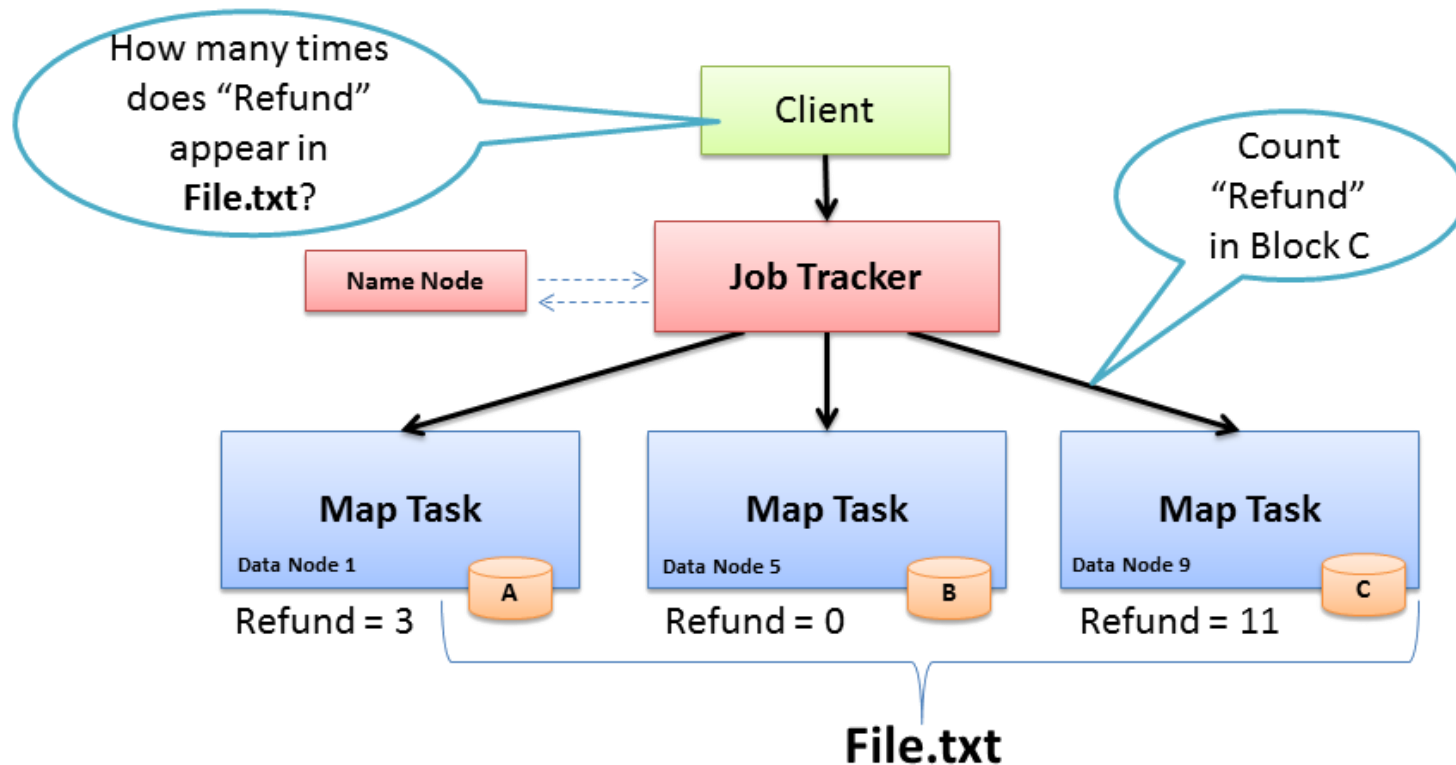


# Apache Hadoop – MapReduce Παράδειγμα



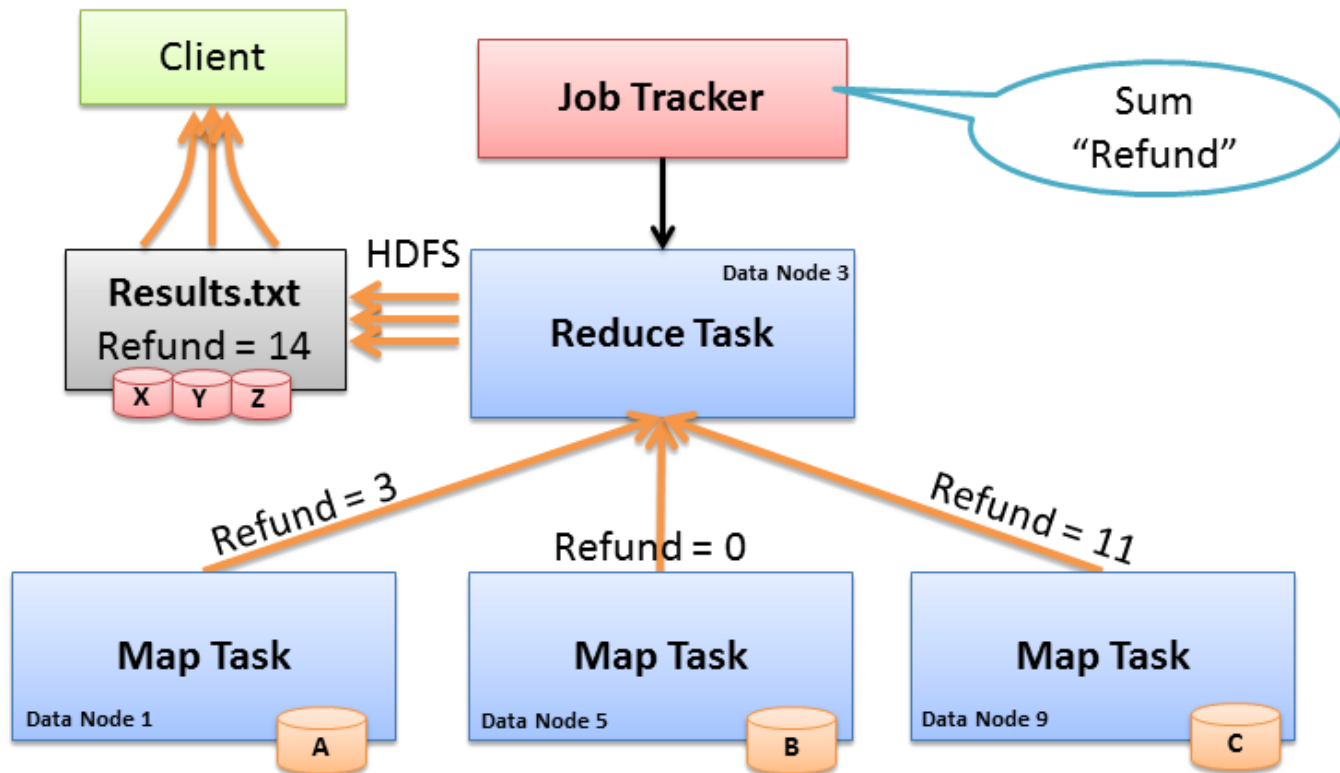
# Apache Hadoop – MapReduce

## Map Phase



# Apache Hadoop – MapReduce

## Reduce Phase



# Apache Hadoop – MapReduce Schedulers

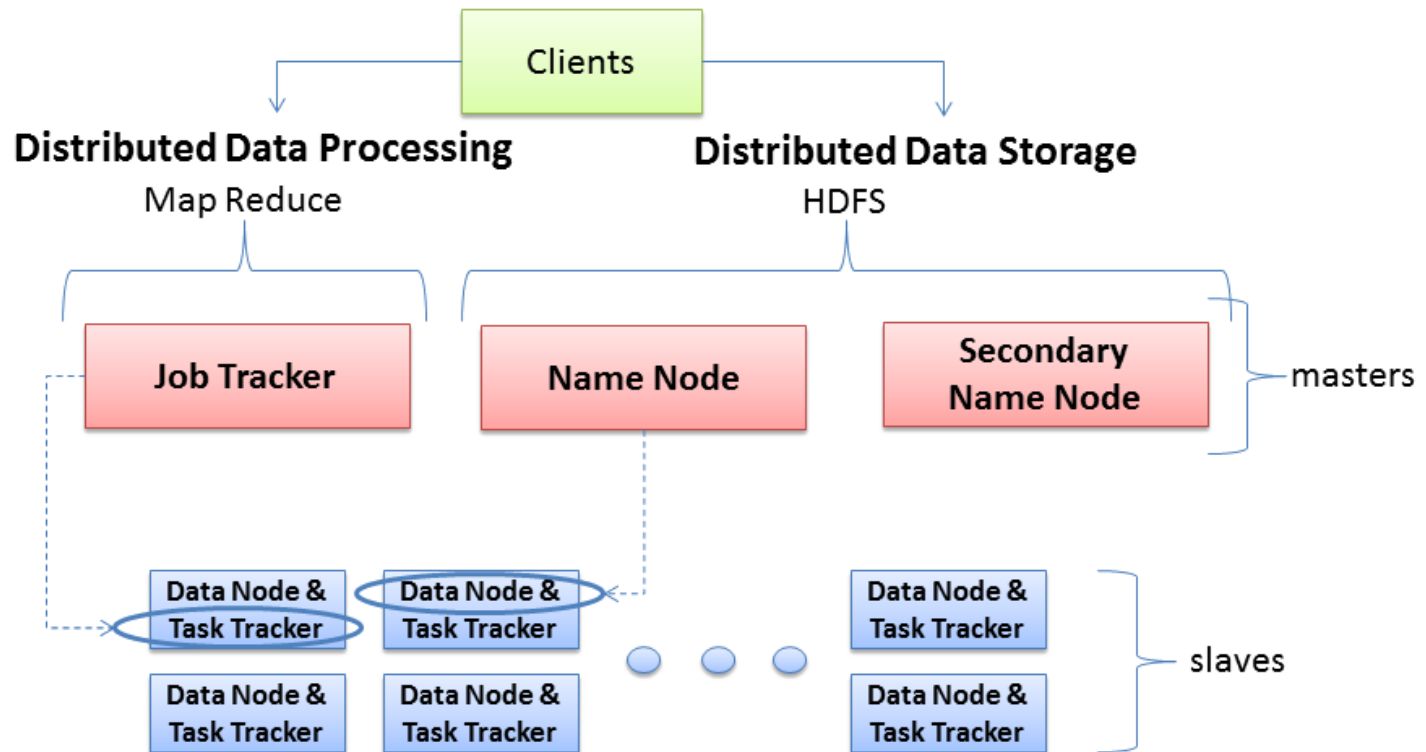
- FIFO
  - Είναι ο προκαθορισμένος scheduler.
- Capacity Scheduler
  - Αναπτύχθηκε από την Yahoo.
  - Οι εργασίες μπαίνουν σε ουρές.
  - Οι εργασίες μπορούν να έχουν προτεραιότητες.
  - Σε κάθε ουρά κατανέμεται ένα μικρό ποσοστό της συνολικών διαθέσιμων πόρων.
  - Οι ελεύθεροι πόροι κατανέμονται στις ουρές που ξεπερνούν την συνολική της χωρητικότητα.
  - Δεν γίνεται να διακοπεί μια εργασία όταν εκτελείται.

# Apache Hadoop – MapReduce Schedulers

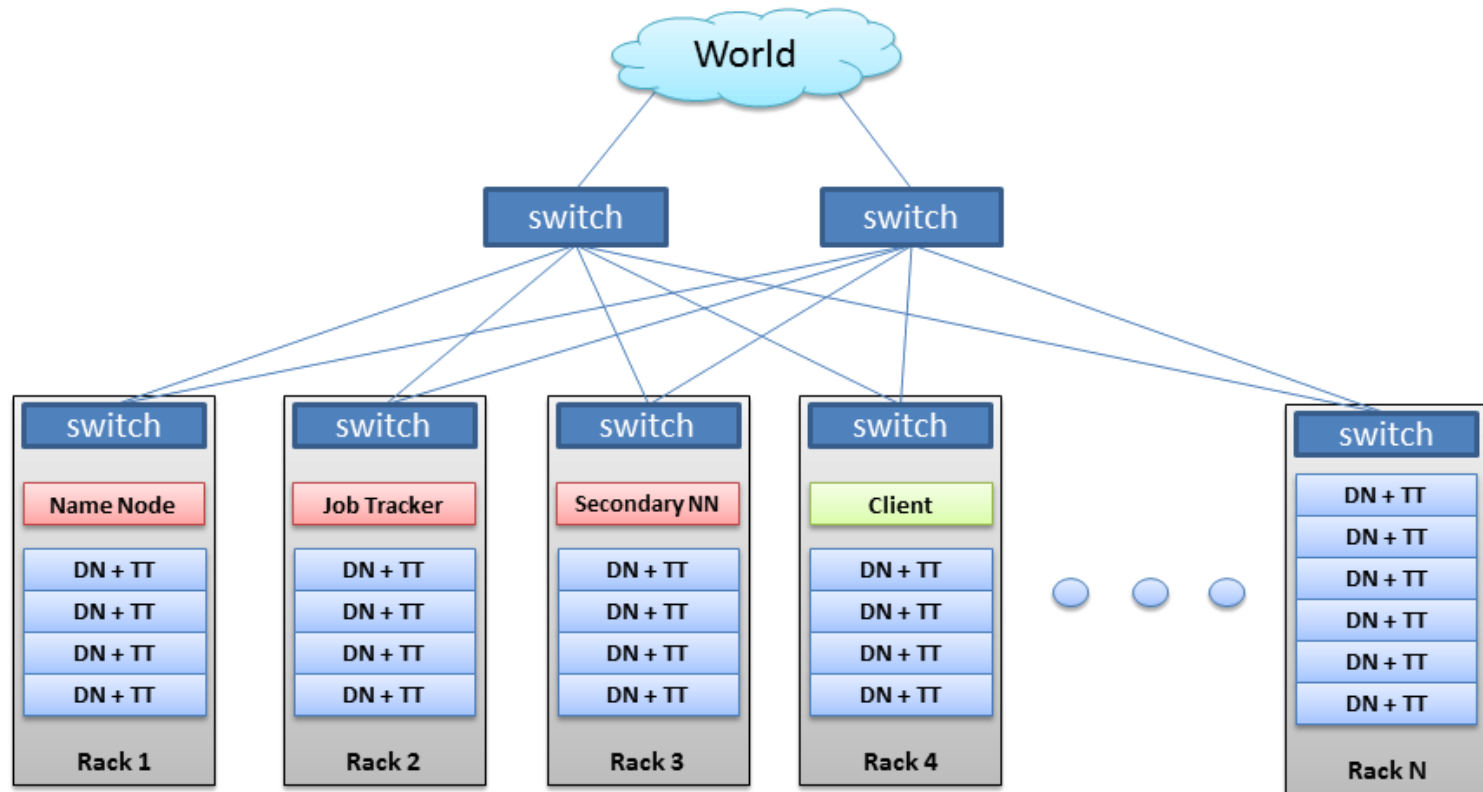
- Fair Scheduler
  - Αναπτύχθηκε από το Facebook.
  - Παρέχει γρήγορους χρόνους απόκρισης για μικρές εργασίες.
  - Οι εργασίες ομαδοποιούνται.
  - Σε κάθε ομάδα κατανέμεται και εγγυάται ένα ελάχιστο ποσοστό των διαθέσιμων πόρων.
  - Η επιπλέον χωρητικότητα μοιράζεται μεταξύ των εργασιών.
  - Οι εργασίες που δεν ανήκουν σε κάποια κατηγορία πάνε σε μια προκαθορισμένη ομάδα.
  - Οι ομάδες πρέπει να καθορίσουν τον ελάχιστο αριθμό map και reduce slot, και ένα όριο στον αριθμό εργασιών.



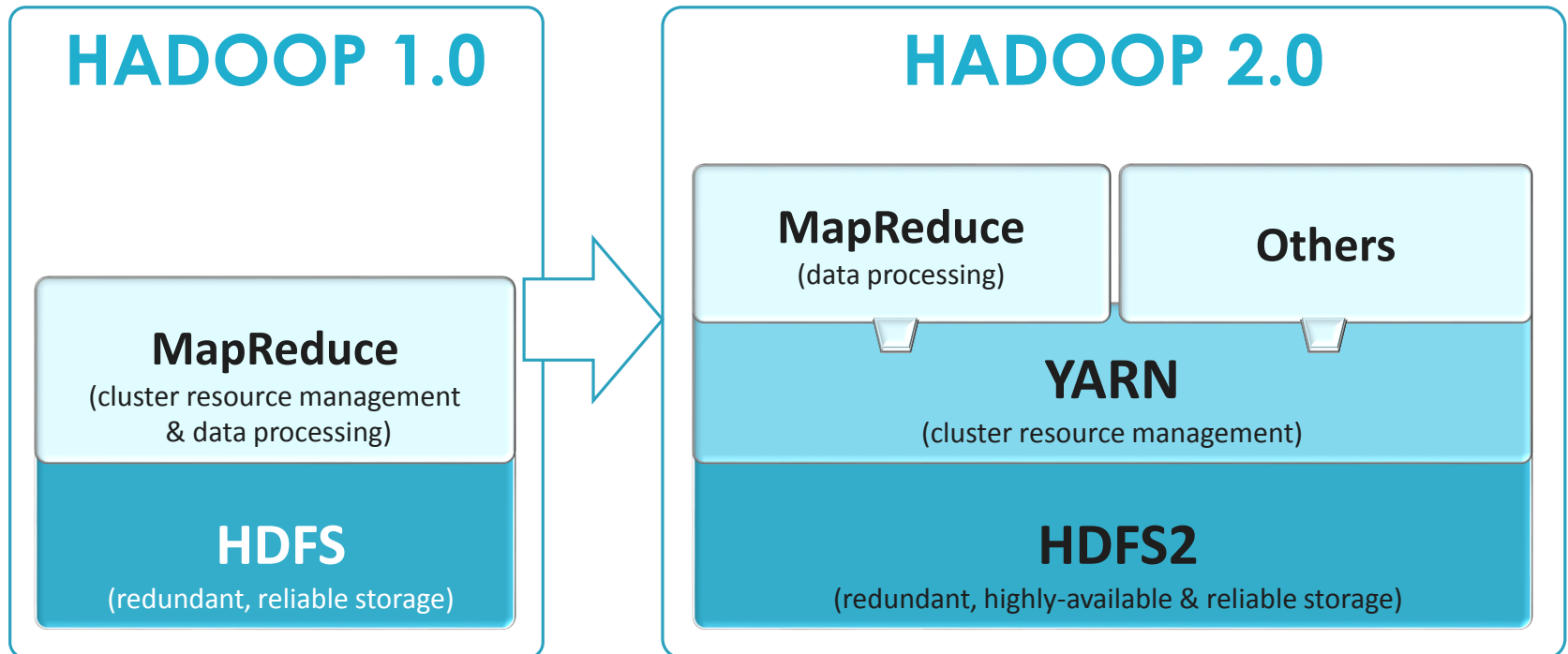
# Apache Hadoop – MapReduce Cluster



# Apache Hadoop – MapReduce Cluster (2)

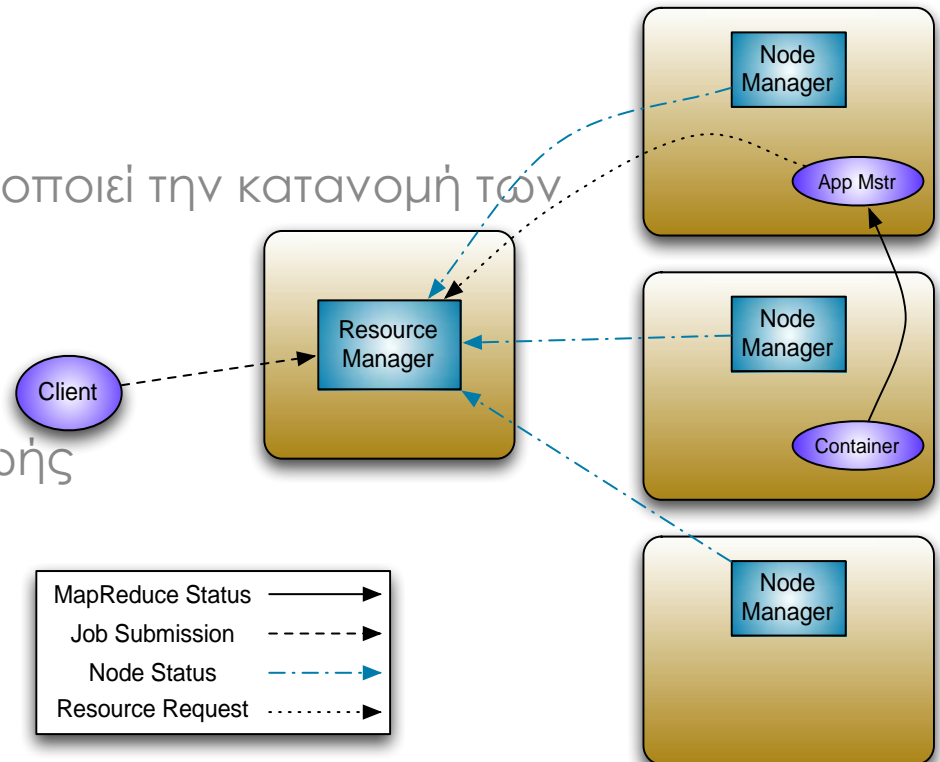


# Apache Hadoop Next Gen – Hadoop YARN



# Apache Hadoop Next Gen – Hadoop YARN

- ResourceManager (RM)
  - Κεντρική διαχείριση και κατανομή των πόρων του cluster.
- NodeManager (NM)
  - Ένας agent ανά κόμβο.
  - Διαχειρίζεται και πραγματοποιεί την κατανομή των πόρων.
- ApplicationMaster (AM)
  - Ένας ανά εφαρμογή.
  - Διαχειρίζεται τον κύκλο ζωής των εφαρμογών και την χρονοδρομολόγηση των εργασιών.



# Apache Hadoop Next Gen – Hadoop YARN

- Πλεονεκτήματα του YARN:
  - Νέες εφαρμογές.
  - Βελτιωμένη αξιοποίηση πόρων.
  - Επεκτασιμότητα.
  - Ταχύτητα και ευκινησία.
  - Νέες υπηρεσίες.

## Applications Run Natively *in* Hadoop

**BATCH**  
(MapReduce)

**INTERACTIVE**  
(Tez)

**ONLINE**  
(HBase)

**STREAMING**  
(Storm, S4,...)

**GRAPH**  
(Giraph)

**IN-MEMORY**  
(Spark)

**HPC MPI**  
(OpenMPI)

**OTHER**  
(Search)  
(Weave...)

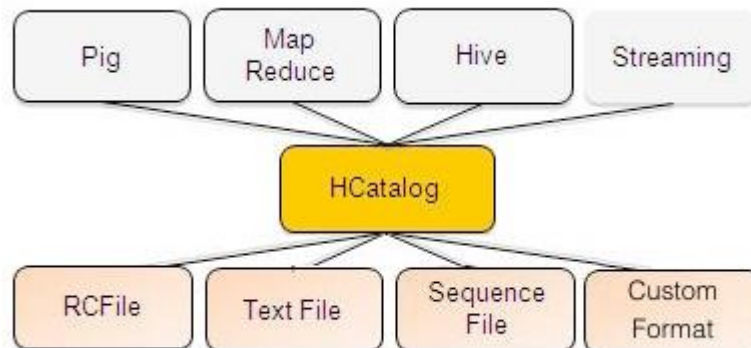
**YARN** (Cluster Resource Management)

**HDFS2** (Redundant, Reliable Storage)



# Apache Hadoop Other Projects

- **Ambari:** ανάπτυξη, ρύθμιση και παρακολούθηση του cluster.
- **Avro:** ένα format για BigData και ένα σύστημα σειριοποίησης των δεδομένων.
- **Cassandra:** Μια επεκτάσιμη βάση δεδομένων, χωρίς μοναδικό σημείο σφάλματος.
- **Chukwa:** Ένα σύστημα συλλογής δεδομένων για την διαχείριση μεγάλων κατανεμημένων συστημάτων.
- **Flume:** Συλλογή και εισαγωγή δεδομένων γεγονότων και καταγραφής.
- **HBase:** Μια column-oriented κατανεμημένη βάση δεδομένων, βασισμένη στο Google Big Table, επεκτάσιμη έως και σε δισεκατομμύρια γραμμών
- **HCatalog:** Ένας μηχανισμός τύπου δεδομένων και ένα κοινό σχήμα για εργαλεία του Hadoop. Παρέχει μία αφηρημένη έννοια του πίνακα προς τους χρήστες και διαλειτουργικότητα για εργαλεία όπως το Pig, το Hive και το MapReduce.



# Apache Hadoop Other Projects (2)

- **Hive:** Αποθήκη δεδομένων για εκτέλεση ερωτημάτων τύπου SQL (Hive QL)
- **Mahout:** Βιβλιοθήκη αλγορίθμων μηχανικής μάθησης και εξόρυξης δεδομένων.
- **Pig:** Υψηλού επιπέδου γλώσσα προγραμματισμού τύπου script για εκτέλεση Hadoop εργασιών.
- **Oozie:** Ένα σύστημα χρονοδρομολόγησης ροών δεδομένων για την διαχείριση Hadoop εργασιών.
- **Sqoop:** Εισαγωγή δεδομένων από σχεσιακές βάσεις δεδομένων.
- **Whirr:** Εργαλείο για ανάπτυξη Hadoop cluster σε cloud πλατφόρμες.
- **Zookeeper:** Κατανεμημένη υπηρεσία διαχείρισης των ρυθμίσεων, του συγχρονισμού και του συντονισμού του cluster.

# Apache Hadoop Other Projects (3)

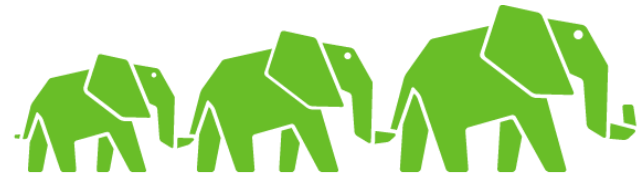
- **Tez** : Γενικεύει το μοντέλο MapReduce για την εκτέλεση εργασιών πολύπλοκων κατευθυνόμενων άκυκλων γράφων.
- **Knox Gateway**: Ένα σύστημα που παρέχει ένα κεντρικό σημείο αυθεντικοποίησης και πρόσβασης για τις υπηρεσίες ενός Hadoop cluster.
- **Falcon**: Ένα framework για την διαχείριση των δεδομένων και την επεξεργασία μέσω πληροφορίας (pipelines).
- **Storm**: Ένα καταναμημένο σύστημα υπολογισμού πραγματικού χρόνου για την γρήγορη επεξεργασία μεγάλων ροών δεδομένων.
- **Accumulo**: Ένα σύστημα αποθήκευσης και ανάκτησης δεδομένων υψηλών επιδόσεων με έλεγχο πρόσβασης σε επίπεδο κελιού.



# Apache Hadoop Distributions



cloudera



Hortonworks



MAPR™  
TECHNOLOGIES

# Apache Hadoop – Machine Learning

- Μάθηση με επίβλεψη (supervised learning) ή μάθηση με παραδείγματα (learning from examples),
- Μάθηση χωρίς επίβλεψη (unsupervised learning) ή μάθηση από παρατήρηση (learning from observation).
- Στην μάθηση με επίβλεψη διακρίνονται δυο είδη προβλημάτων (learning tasks), τα προβλήματα ταξινόμησης και τα προβλήματα παρεμβολής.
  - Η ταξινόμηση (classification) αφορά στη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων (κλάσεων/κατηγοριών) (π.χ. ομάδα αίματος).
  - Η παρεμβολή (regression) αφορά στη δημιουργία μοντέλων πρόβλεψης αριθμητικών τιμών (π.χ. πρόβλεψη ισοτιμίας νομισμάτων ή τιμής μετοχής).

# Apache Hadoop – Machine Learning

- Η ταξινόμηση (classification) είναι μία τεχνική της εξόρυξης δεδομένων, κατά την οποία ένα στοιχείο ανατίθεται σε ένα προκαθορισμένο σύνολο κατηγοριών.
- Η ταξινόμηση μπορεί να περιγραφεί ως μία διαδικασία δύο βημάτων:
  - **Εκμάθηση (Learning):** Στο πρώτο βήμα της διαδικασίας δημιουργείται/προσδιορίζεται το μοντέλο με βάση ένα σύνολο προκατηγοριοποιημένων παραδειγμάτων, που ονομάζεται δεδομένα εκπαίδευσης (training data). Τα δεδομένα εκπαίδευσης αναλύονται από ένα αλγόριθμο κατηγοριοποίησης, προκειμένου να σχηματιστεί το μοντέλο.
  - **Κατηγοριοποίηση (Classification):** Μετά την δημιουργία του μοντέλου, το επόμενο βήμα είναι η αξιολόγησή του. Για να επιτευχθεί αυτό, χρησιμοποιούμε τα δοκιμαστικά δεδομένα (test data) για να υπολογίσουν την ακρίβεια του μοντέλου. Το μοντέλο κατηγοριοποιεί τα δοκιμαστικά δεδομένα.

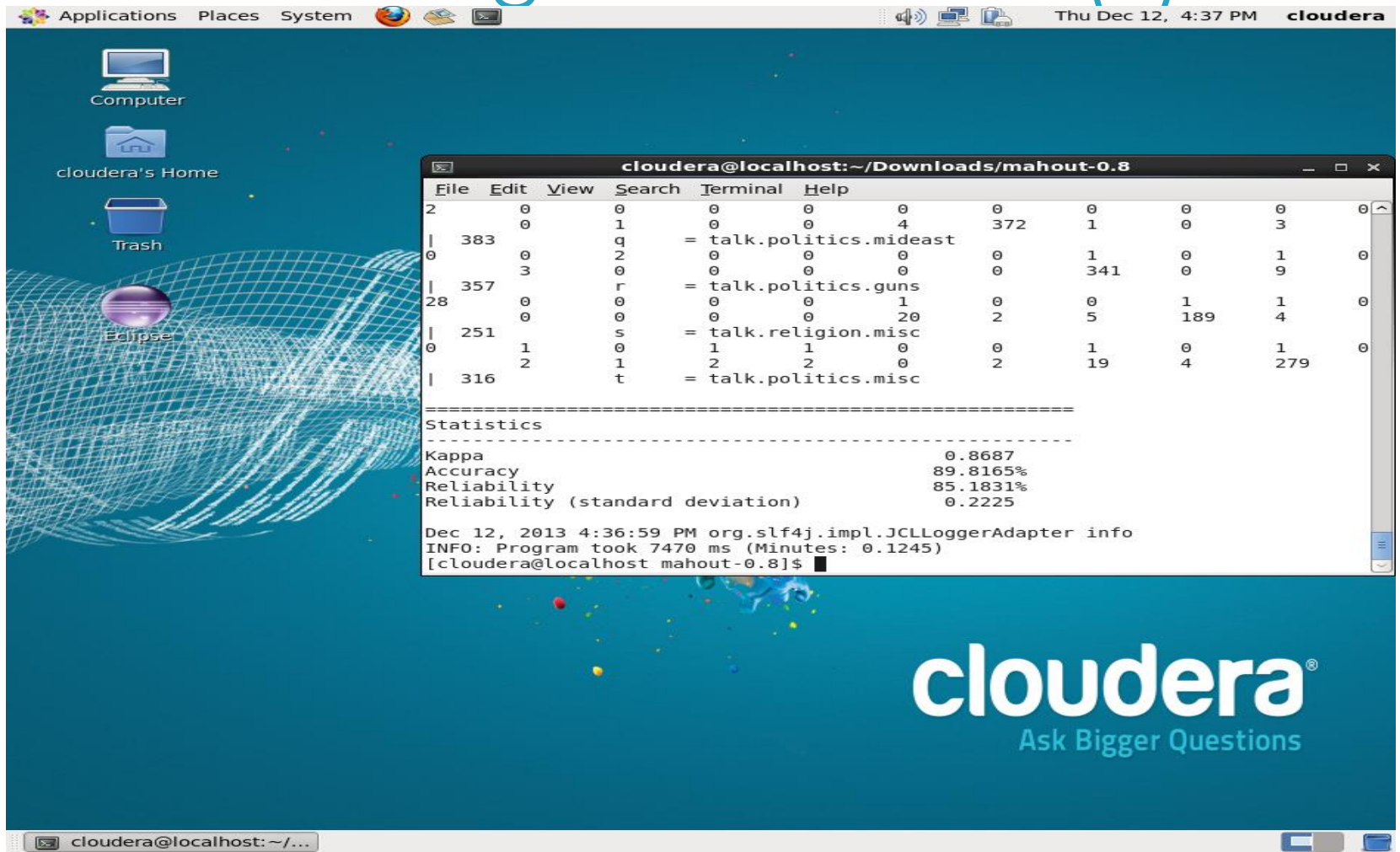
# Apache Hadoop – Machine Learning

- Η Bayesian ταξινόμηση αποτελεί μία κατηγορία μεθόδων της ταξινόμησης και βασίζεται στη στατιστική θεωρία κατηγοριοποίησης του Bayes. Αυτό σημαίνει ότι πραγματοποιείται μια πιθανοτική πρόβλεψη, δηλαδή προβλέπει την πιθανότητα ένα δείγμα  $X$  να ανήκει σε κάποια κατηγορία.
- Ο απλούστερος Bayesian ταξινομητής είναι ο Naïve Bayesian, στον οποίο γίνεται η παραδοχή ότι τα χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους.
- $p(C_i | x) > p(C_j | x)$  για κάθε  $1 \leq j \leq m$  και  $j \neq i$
- Ο στόχος, λοιπόν, είναι να βρούμε το μέγιστο  $p(C_i | X)$  για κάθε κλάση, με αποτέλεσμα ο Naïve Bayesian ταξινομητής να έχει υψηλή απόδοση.

# Apache Hadoop – Machine Learning with Mahout

- πολυπλοκότητα Naïve Bayes  $\rightarrow O(n*m)$
- Mahout κατάλληλο για τέτοιου είδους προβλήματα
- Παράδειγμα Mahout:
  - Twenty Newsgroup dataset – 20,000 έγγραφα, διαχωρισμένα σε 20 ομάδες.
  - Εκτέλεση παραδείγματος σε:
    - Oracle VM Virtualbox 4.3
    - CentOS 6 x64bit
    - Ssh-server
    - Cloudera CDH 4.4
    - Apache Maven 3.0.4
    - Apache Mahout 0.8

# Apache Hadoop – Machine Learning with Mahout (2)



The screenshot shows a Linux desktop environment with a blue background. The desktop has icons for 'Computer', 'cloudera's Home', 'Trash', and 'Eclipse'. A terminal window is open, displaying the output of a Mahout command. The terminal title is 'cloudera@localhost:~/Downloads/mahout-0.8'. The output shows a list of terms and their associated counts, followed by a 'Statistics' section.

```
cloudera@localhost:~/Downloads/mahout-0.8
File Edit View Search Terminal Help
2 0 0 0 0 0 0 0 0 0
| 383 0 1 0 0 4 372 1 0 3
0 0 2 0 0 0 0 1 0 1
| 357 3 0 0 0 0 0 341 0 9
28 0 0 0 0 1 0 0 1 1
| 251 0 0 0 0 20 2 5 189 4
0 1 0 1 1 0 0 1 0 1
| 316 2 1 2 2 0 2 19 4 279

=====
Statistics
-----
Kappa 0.8687
Accuracy 89.8165%
Reliability 85.1831%
Reliability (standard deviation) 0.2225

Dec 12, 2013 4:36:59 PM org.slf4j.impl.JCLLoggerAdapter info
INFO: Program took 7470 ms (Minutes: 0.1245)
[cloudera@localhost mahout-0.8]$
```

cloudera®  
Ask Bigger Questions

cloudera@localhost:~/...



Ευχαριστώ!