

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΙΣΧΥΟΣ ΥΠΟΛΟΓΙΣΜΩΝ ΣΥΣΤΟΙΧΙΑΣ: ΕΝΑ ΠΕΙΡΑΜΑ ΜΕ HADOOP

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΠΜΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ,
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΧΑΝΤΖΟΠΛΑΚΗ ANNA

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΜΑΡΓΑΡΙΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ



Στόχοι

- Κατανόηση λειτουργίας της πλατφόρμας Hadoop και του μοντέλου MapReduce
- Πρόταση μοντέλου μέτρησης ισχύος χωρίς τη χρήση εξωτερικού εξοπλισμού
- Αξιολόγηση επιδόσεων εφαρμογών
- Αξιολόγηση κατανάλωσης ισχύος και ενεργειακής αποδοτικότητας των εφαρμογών κάτω από διαφορετικές συνθήκες

Η Πλατφόρμα Hadoop

- Το Hadoop είναι ένα ανοιχτού κώδικα λογισμικό (framework) που υποστηρίζει την παράλληλη επεξεργασία μεγάλου όγκου δεδομένων διανέμοντας τμήματα της εφαρμογής σε μεγάλες συστάδες (clusters) υπολογιστών.
- Είναι γραμμένο σε Java.
- Σκοπός του είναι να παρέχει αξιόπιστη και ταχύτατη ανάλυση πληροφοριών από και προς τις εφαρμογές.
- Πλεονεκτήματα: επεκτάσιμο, fault tolerant, τρέχει σε συμβατικό hardware.
- Το Hadoop βασίστηκε στο Google Map Reduce framework (2004) και το Google File System (GFS). Είναι ένα έργο του Apache Software Foundation.
- Χρησιμοποιείται από: Google, Yahoo, Facebook, Amazon.

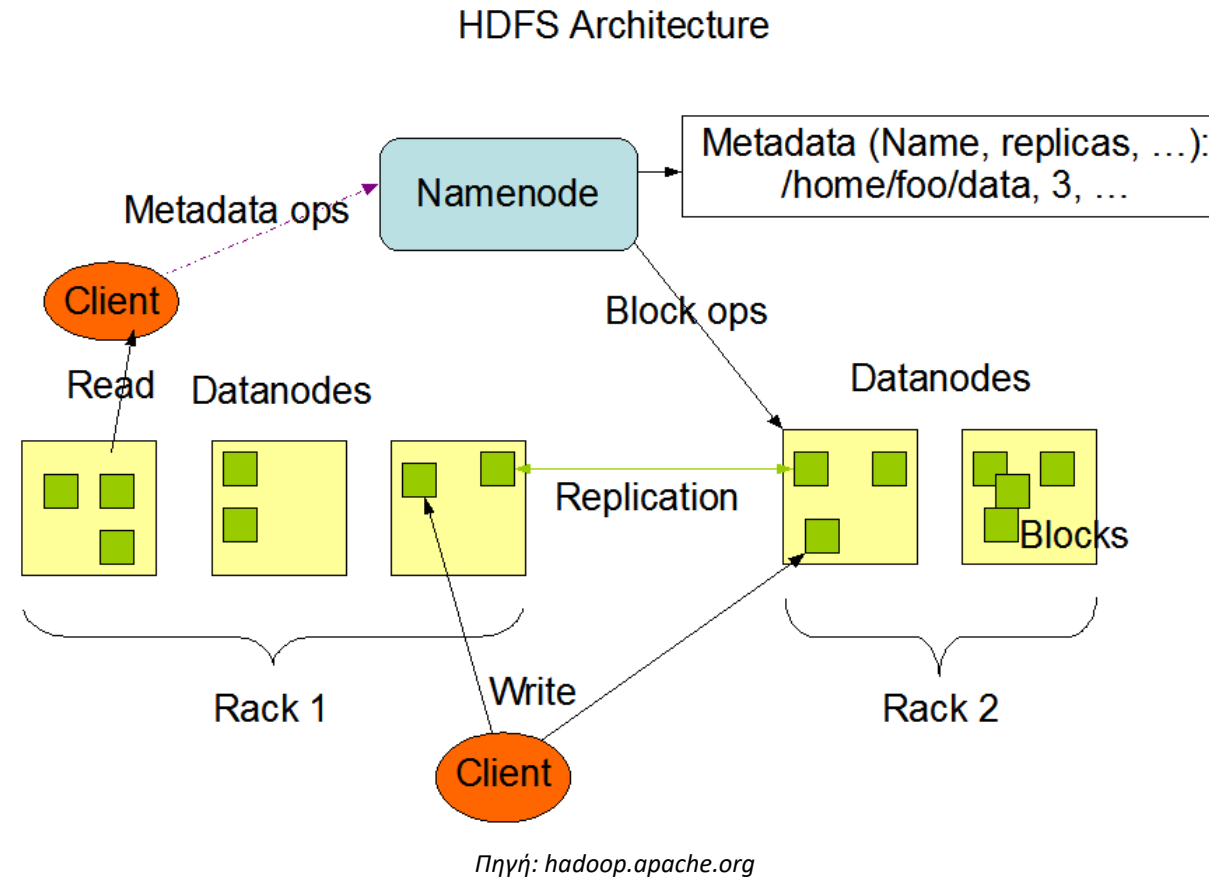
Η Πλατφόρμα Hadoop

- Παρέχει δύο βασικές υπηρεσίες (υλοποιήσεις):
 - την αξιόπιστη αποθήκευση δεδομένων με χρήση του Hadoop Distributed File System (HDFS)
 - και την ταχύτατη, παράλληλη, επεξεργασία δεδομένων χρησιμοποιώντας μια τεχνική με την ονομασία MapReduce.

Κατανεμημένο Σύστημα Αρχείων Hadoop

- Αρχιτεκτονική: master/slave.
 - Έναν κεντρικό κόμβο, τον NameNode, που κρατά πληροφορίες για το που βρίσκονται τα δεδομένα στο HDFS (namespace, block location).
 - Secondary NameNode, ο οποίος κρατά snapshots των φακέλων του NameNode και μαζί με τα αρχεία ιστορικού του (logs), επαναφέρει το σύστημα αρχείων μετά από αποτυχία.
 - DataNodes που αποθηκεύουν τα δεδομένα.
- Οργάνωση δεδομένων:
 - Υποστηρίζει μία παραδοσιακή ιεραρχική οργάνωση αρχείων.
 - Σχεδιάστηκε για να χειρίζεται πολύ μεγάλα αρχεία.
 - Τα αρχεία διασπώνται σε blocks.
 - Τυπικό μέγεθος block 64 MB (fast streaming reads) ή 128MB.
 - Replication: Κάθε block αντιγράφεται σε πολλαπλούς κόμβους δεδομένων (DataNodes) - default 3 (rack aware).

Αρχιτεκτονική HDFS



MapReduce framework

- Ο πυρήνας του Hadoop.
- Το πρόγραμμα εκτελεί δύο διακριτές διεργασίες:

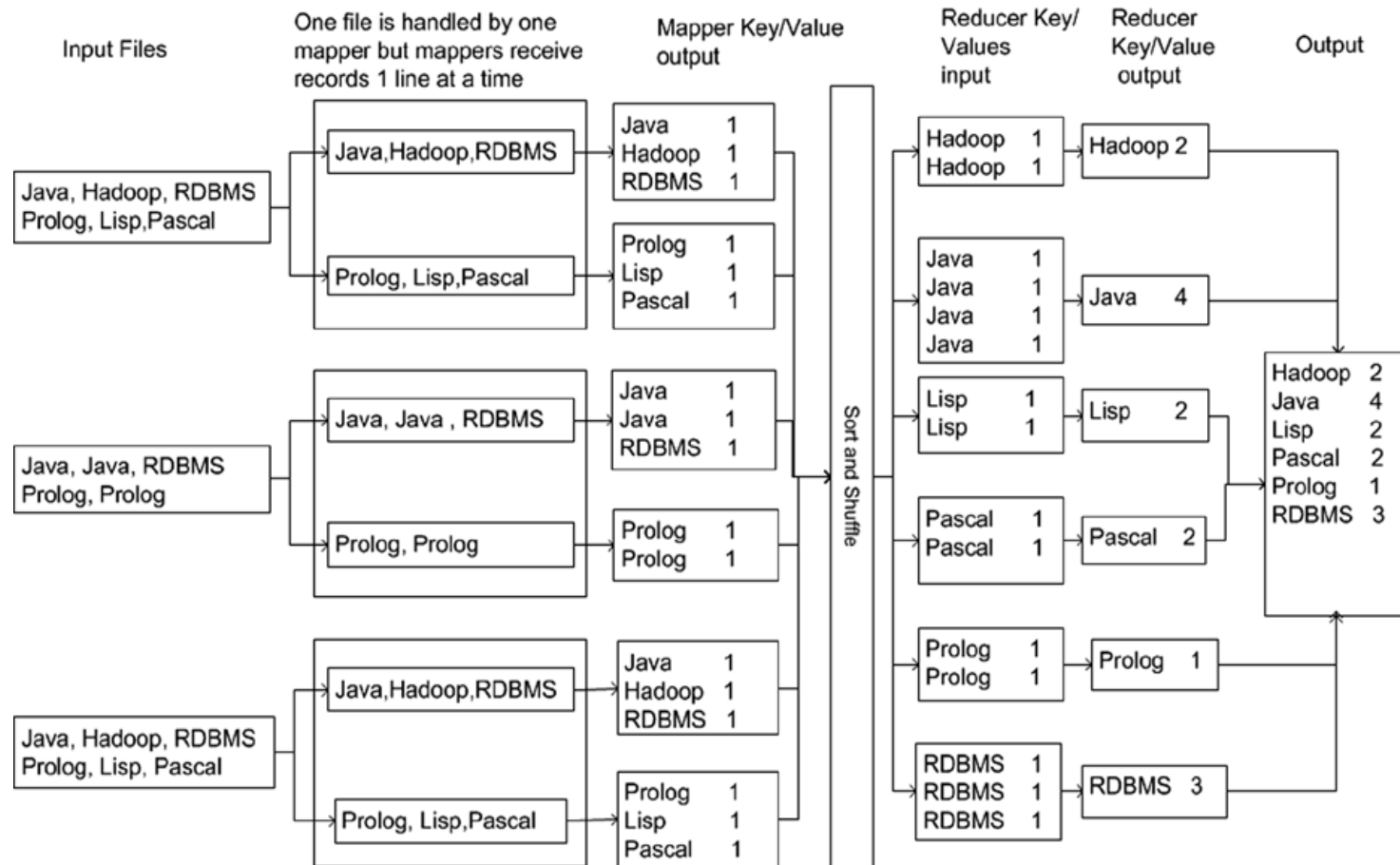
Map

- δέχεται ένα block αρχείου
- το χωρίζει σε ζεύγη <key,value>

Reduce

- δέχεται σαν είσοδο την προηγούμενη έξοδο και την ταξινομεί
- shuffle -> sort -> reduce

WordCount application

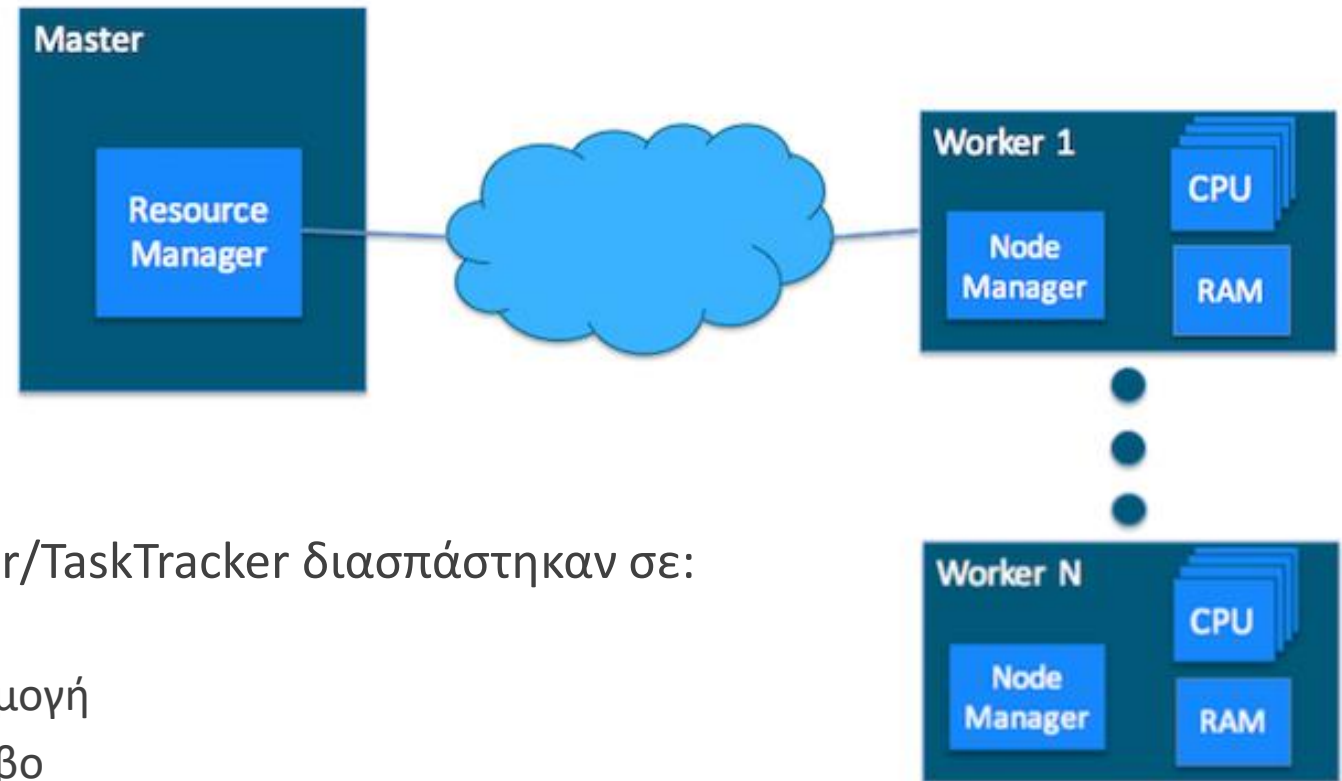


Πηγή: Pro Apache Hadoop

- Μέτρηση εμφάνισης λέξεων σε ένα κείμενο.
- Map task: $\langle \{Word\}, 1 \rangle$
- Reduce task: $\langle \{Word\}, [1, \dots, 1] \rangle$
- Output: $\langle \{Word\}, \{count\ of\ word\} \rangle$

Hadoop YARN Framework

- Ενσωματώθηκε στο Hadoop 2
- Σύστημα Διαχείρισης Πόρων
- Οι προηγούμενες εργασίες των JobTracker/TaskTracker διασπάστηκαν σε:
 - έναν καθολικό ResourceManager (RM)
 - έναν ApplicationMaster (AM) για κάθε εφαρμογή
 - έναν NodeManager (NM) για κάθε slave κόμβο
 - ένα container για κάθε εφαρμογή που τρέχει σε έναν NM



Πηγή: blog.cloudera.com

Hadoop YARN Framework

1

- Υποβολή εφαρμογής

2

- RM παραχωρεί container στον AM

3

- AM ζητάει περισσότερα containers από τον RM
- Συνεργάζεται με τον NM που παρακολουθεί τις εκτελέσεις των containers

4

- Όταν ολοκληρωθεί η εφαρμογή ο AM τερματίζει τη λειτουργία του

Υλοποίηση Πειράματος

- Cluster 7 quad-core κόμβων, Ubuntu 16.04.1
- Εγκατάσταση Hadoop version 2.7.2
- Εγκατάσταση Ganglia monitoring tool
- 1 Power meter συνεδεδεμένο με τον κεντρικό κόμβο.

Μοντέλο Πρόβλεψης Ισχύος

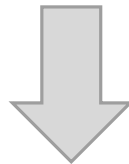
- Βασίζεται στο paper των *Bohra & Chaudhary*
- Ισχυρή συσχέτιση μεταξύ της ισχύς και της χρήσης των μετρικών: CPU, cache, memory, disk
- Προτείνουν το παρακάτω γραμμικό μοντέλο:

$$P(total) = c_0 + c_1 p_{CPU} + c_2 p_{cache} + c_3 p_{DRAM} + c_4 p_{disk}$$

Μοντέλο Πρόβλεψης Ισχύος: Υπολογισμός

- Χρήση πραγματικών μετρήσεων του Power Meter
- Script για την εξαγωγή των logs των μετρικών από τα Ganglia και τον υπολογισμό των ποσοστών χρήσης αυτών
- Δημιουργία 60 εξισώσεων της παρακάτω μορφής και μετατροπή σε μήτρες
- Χρήση Γραμμικών Ελάχιστων Τετραγώνων (LLS): $A * c = b$

$$198.76462 = 165 + c_1 * 0.696307 + c_2 * 0.427565 + c_3 * 0.707971 + c_4 * 0.197646$$



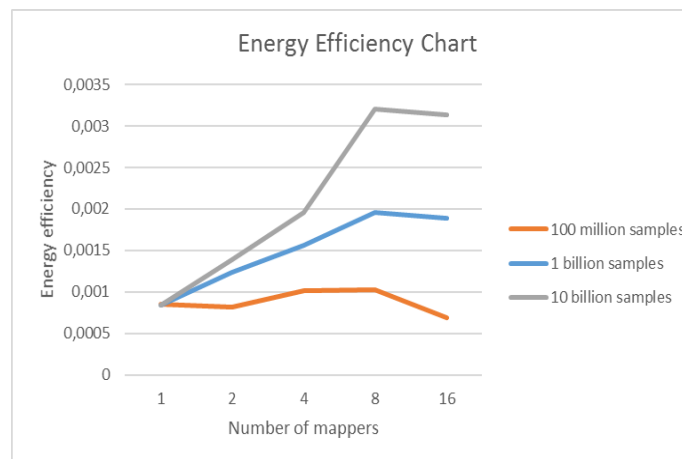
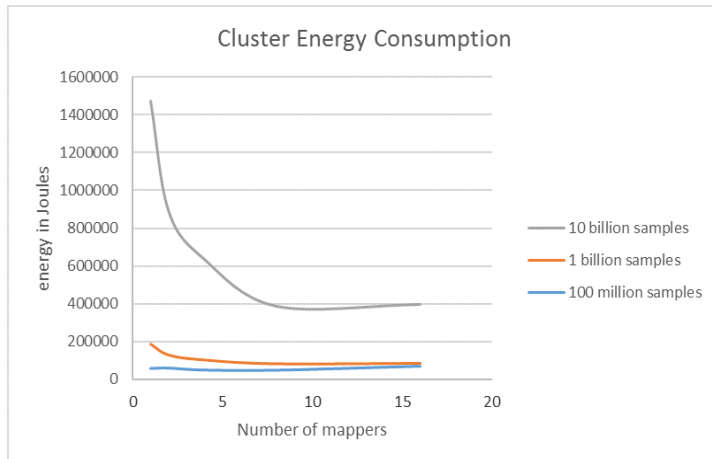
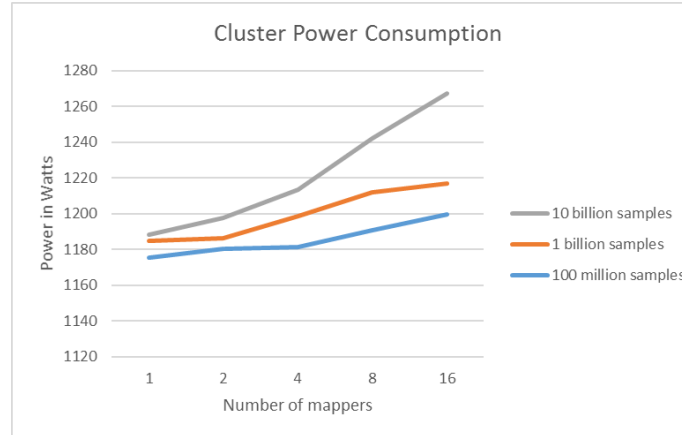
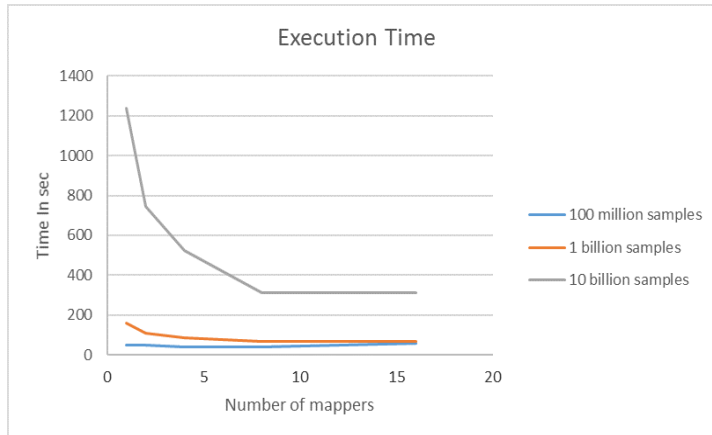
$$P(total) = 165 - 21.834 * p_{CPU} - 38.706 * p_{cache} + 97.267 * p_{DRAM} - 37.389 * p_{disk}$$

Μοντέλο Πρόβλεψης Ισχύος: Σταθερότητα

- Σχετικό σφάλμα κυμαίνεται από -9% έως +9%.

Power estimated	Power measured	Relative error %
219.7697143555	214.0643466667	-2.6652582635
215.6363220215	218.8448145455	1.4661039745
218.4002990723	225.3061918182	3.065114496
218.8817596436	220.3330018182	0.6586585589
218.8940124512	224.3009163636	2.4105581021

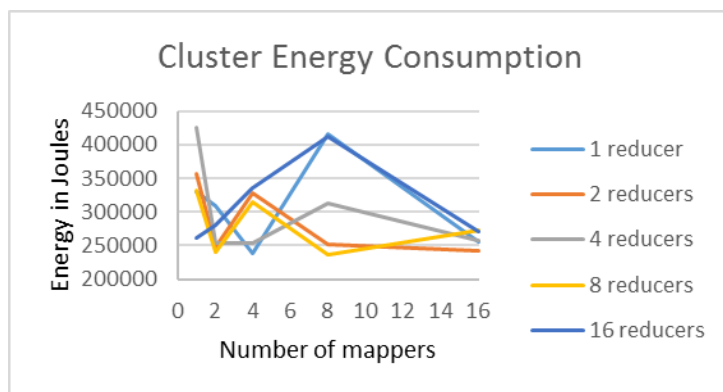
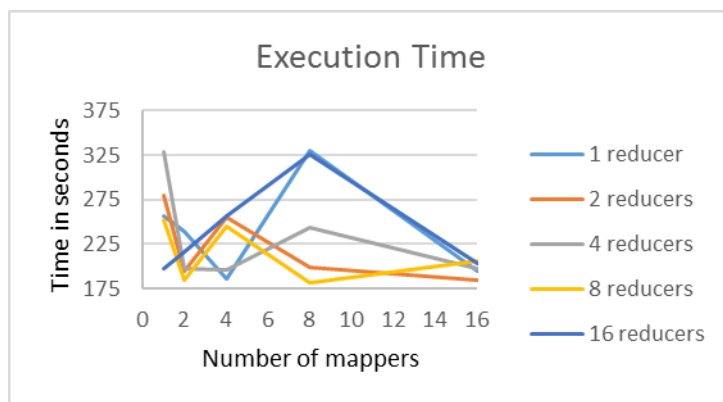
Benchmarks tested: 1. Pi calculation



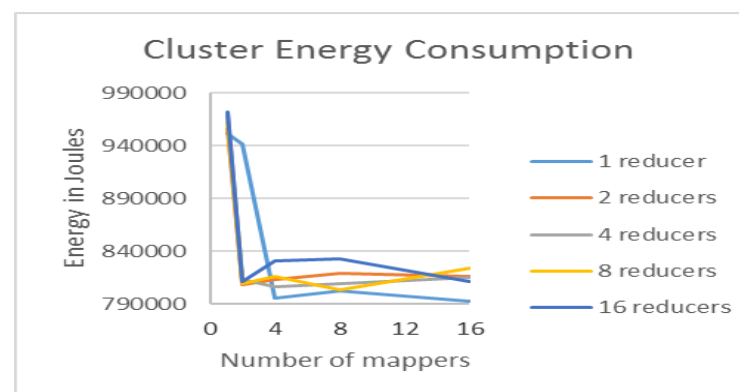
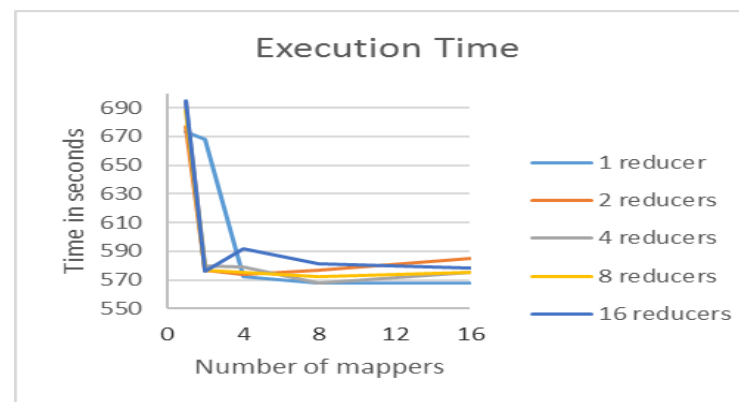
- CPU-intensive
- Embarrassingly parallel
- Παραλληλοποίηση σε μικρό μέγεθος υπολογισμών: μεγάλο overhead
- Παραλληλοποίηση σε μεγάλο μέγεθος υπολογισμών: μείωση χρόνου και αύξηση ενεργειακής αποδοτικότητας

Benchmarks tested: 2. WordCount 1GB

1 input file: 8 splits

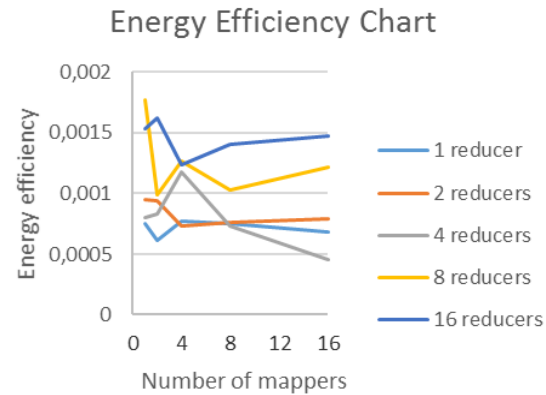
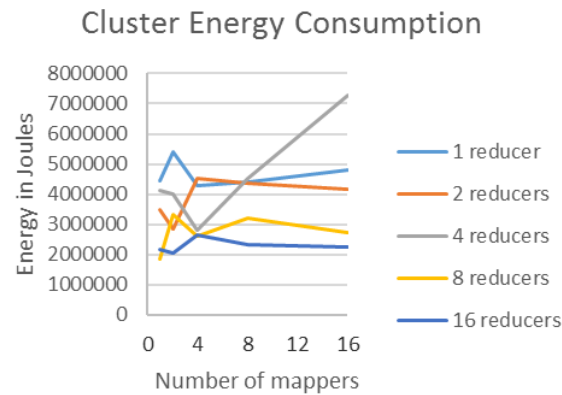
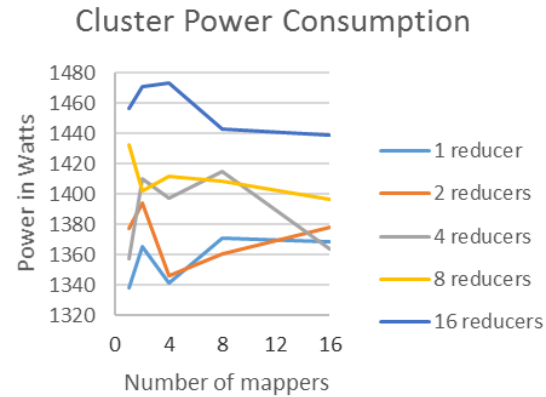
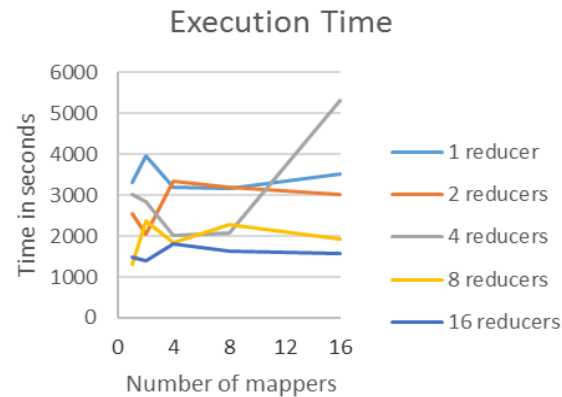


Many input file: 505 splits



- CPU-intensive
- Πολλά μικρά αρχεία είναι μη αποδοτικό
- 1 μεγάλο αρχείο:
 - ✓ Μείωση του χρόνου στο μισό
 - ✓ Σημαντική διαφορά στην κατανάλωση ενέργειας
- Καλύτερη επίδοση: 8mappers-8reducers

Benchmarks tested: 3. Terasort 10GB

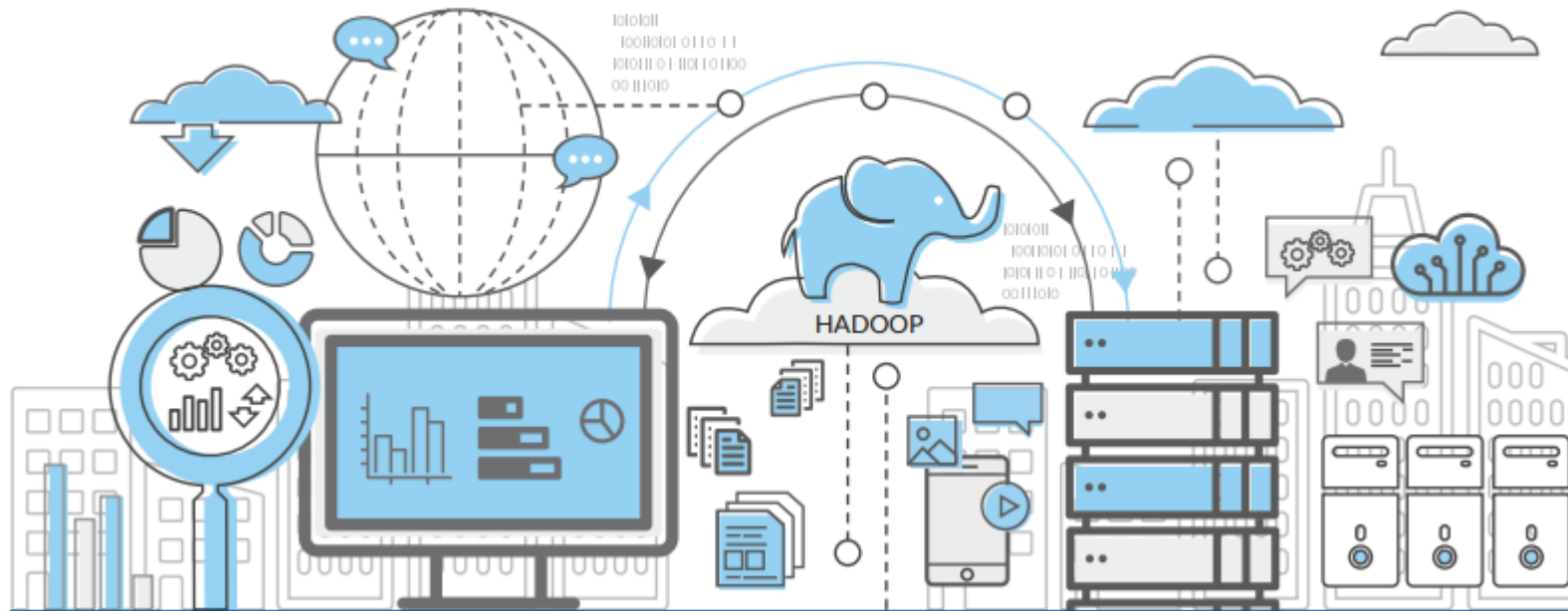


- I/O-intensive
- 76 input splits of 128MB
- Καλύτερη επίδοση: 16 reducers
- Shuffle phase -> bottleneck: μεταφορά μεγάλου όγκου δεδομένων μεταξύ των hosts

Συμπεράσματα

Η απόδοση των εφαρμογών MapReduce εξαρτώνται από:

1. Μέγεθος των υπολογισμών
2. Μέγεθος των αρχείων εισόδου
3. Αριθμός των mappers/reducers ανάλογα με τις ανάγκες της κάθε εφαρμογής



Πηγή: www.talentica.com

Ευχαριστώ!