



Ανάκτηση Πληροφορίας

Ενότητα 10: Παράλληλη Ανάκτηση

Απόστολος Παπαδόπουλος
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





Παράλληλη Ανάκτηση Πληροφορίας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα ενότητας

1. Παράλληλη Ανάκτηση Πληροφορίας





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Παράλληλη Ανάκτηση Πληροφορίας

Εισαγωγή

- Ο όγκος των δεδομένων αυξάνεται με πολύ γρήγορους ρυθμούς.
- Η απόδοση ενός συστήματος IR εξαρτάται άμεσα από τον αριθμό των εγγράφων που αυτό διαχειρίζεται και από το πλήθος και την πολυπλοκότητα των ερωτημάτων (queries).
- Η απόδοση ενός συστήματος IR βελτιώνεται με τη χρήση **παράλληλων ή κατανεμημένων αρχιτεκτονικών**.



Παράλληλα Συστήματα-1

- **Παράλληλη Επεξεργασία:** η ταυτόχρονη λειτουργία πολλών πόρων (επεξεργαστών ή μονάδων I/O) για την επίλυση ενός προβλήματος.
- Η χρήση πολλών πόρων μπορεί να βελτιώσει σημαντικά το χρόνο επεξεργασίας ενός ερωτήματος, διότι η κάθε μονάδα συνεισφέρει στην επίλυση ενός μέρους του αρχικού προβλήματος.



Παράλληλα Συστήματα-2

- Παραλληλισμός μπορεί να εφαρμοστεί σε όλους τους πόρους ενός συστήματος.
- Συνήθως αναφερόμαστε σε :
 - παραλληλισμό επεξεργαστών (processor ή CPU parallelism),
 - παραλληλισμό συστήματος εισόδου-εξόδου (I/O parallelism).



Παραλληλισμός Επεξεργαστών

Ταξινόμηση αρχιτεκτονικών κατά Flynn:

SISD: single instruction stream, single data stream.

SIMD: single instruction stream, multiple data stream.

MISD: multiple instruction stream, single data stream.

MIMD: multiple instruction stream, multiple data stream.

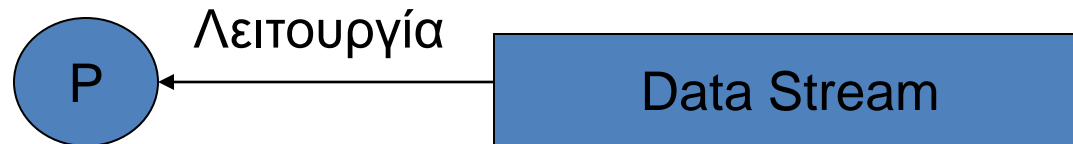


SISD

- Αποτελεί την κλασική μηχανή von Neumann, όπου έχουμε μόνο έναν επεξεργαστή ο οποίος εκτελεί μία ακολουθία εντολών σε ένα stream δεδομένων.
- Στην αρχιτεκτονική SISD στηρίζεται η λειτουργία των προσωπικών υπολογιστών (PC).



SISD Παράδειγμα

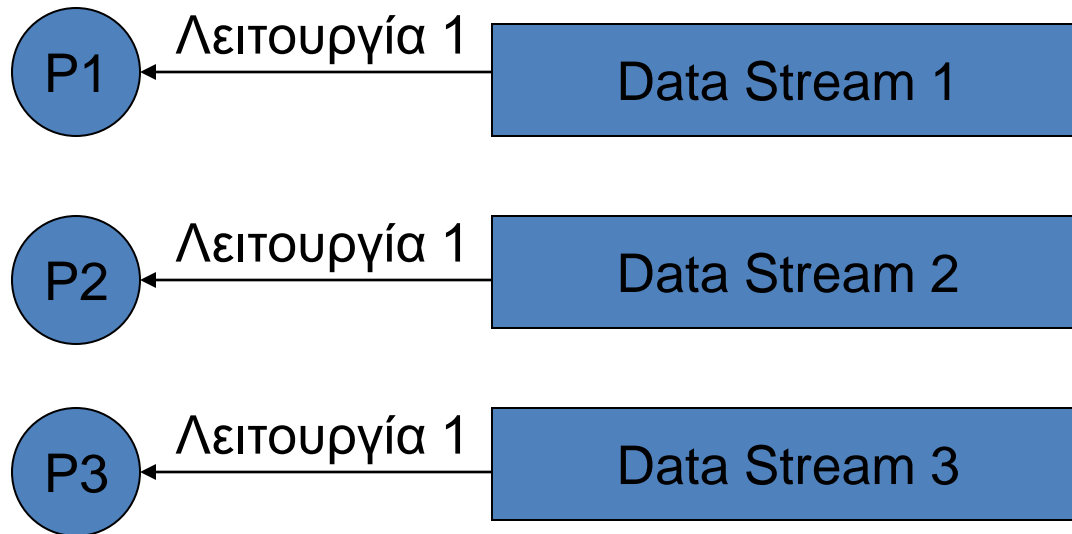


SIMD

- Αποτελείται από N επεξεργαστές οι οποίοι επεξεργάζονται N διαφορετικά δεδομένα εφαρμόζοντας την ίδια λειτουργία.
- Οι επεξεργαστές είναι συγχρονισμένοι ώστε να εκτελούν την ίδια λειτουργία στα δεδομένα.
- Παράδειγμα SIMD συστήματος: Thinking Machines CM-2.
- Κάθε επεξεργαστής μπορεί να έχει τη δική του μνήμη ή όλοι επεξεργαστές μπορούν να μοιράζονται την ίδια μνήμη.



SIMD Παράδειγμα

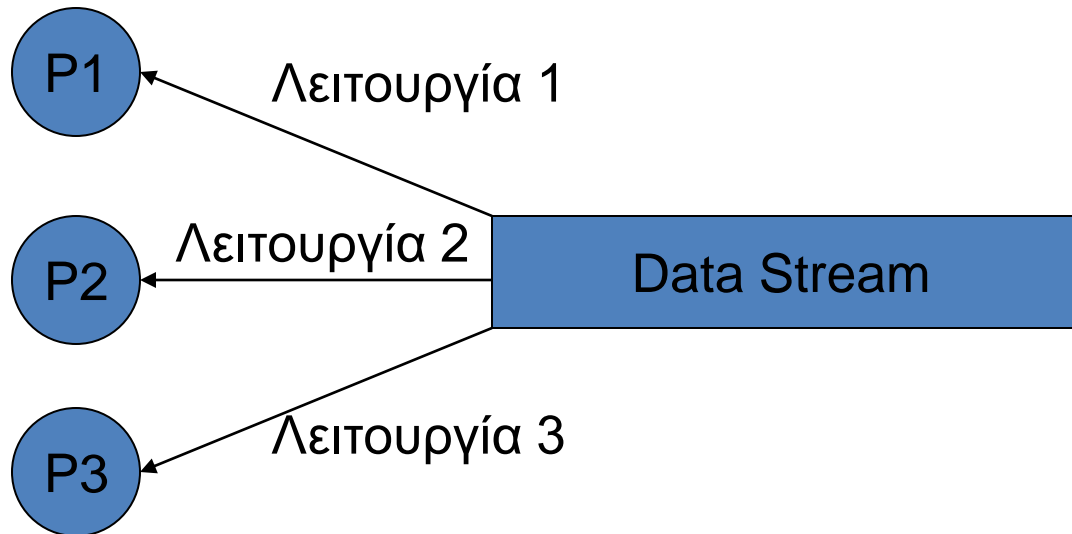


MISD

- N επεξεργαστές οι οποίοι μπορούν να εκτελούν διαφορετικές λειτουργίες στα ίδια δεδομένα.
- Όλοι οι επεξεργαστές μοιράζονται την ίδια μνήμη.
- Τα συστήματα MISD δεν είναι ευρέως διαδεδομένα.
- Παράδειγμα: Systolic Arrays



MISD Παράδειγμα



MIMD-1

- Είναι η πιο γενική και πιο διαδεδομένη παράλληλη αρχιτεκτονική συστημάτων.
- Έχουμε N επεξεργαστές και N δεδομένα, και κάθε επεξεργαστής μπορεί να εκτελεί διαφορετική λειτουργία.
- Κάθε επεξεργαστής μπορεί να έχει τη δική του μνήμη ή όλοι να μοιράζονται την ίδια.
- Παραδείγματα: Sun HPC Server, IBM SP2



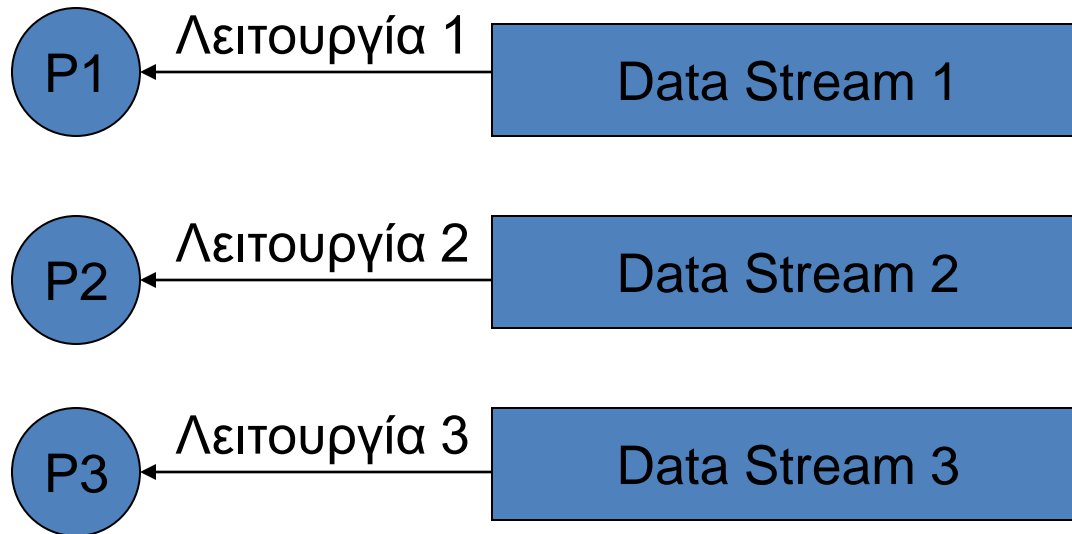
MIMD-2

Ανάλογα με το βαθμό επικοινωνίας των επεξεργαστών μεταξύ τους διακρίνουμε:

- Tightly-coupled συστήματα.
- Loosely-coupled συστήματα.



MIMD Παράδειγμα

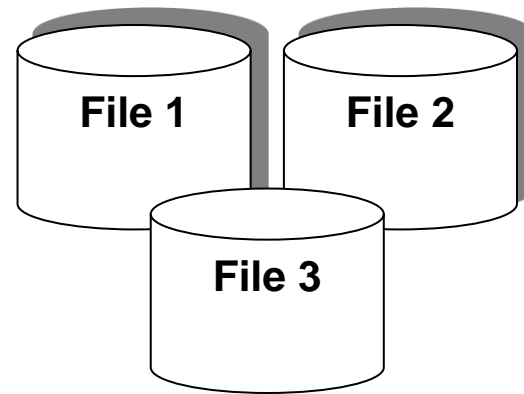
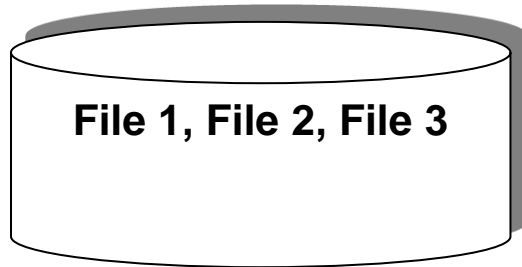


Παραλληλισμός I/O-1

- Κατά τη διάρκεια επεξεργασίας ενός ερωτήματος πολύς χρόνος δαπανάται για λειτουργίες I/O.
- Θέλουμε οι λειτουργίες αυτές να εκτελούνται γρηγορότερα.
- Λύσεις:
 - Γρηγορότερο hardware
 - Περισσότερη RAM
 - Χρησιμοποίηση πολλών μονάδων I/O



Παραλληλισμός I/O-2



Παραλληλισμός I/O-3

- Ο τρόπος με τον οποίο διαμοιράζονται τα δεδομένα στους δίσκους παίζει σημαντικό ρόλο στην επίδοση του συστήματος.
- Ο διαμοιρασμός (declustering) μπορεί να πραγματοποιηθεί:
 - από το υλικό (π.χ. I/O controller)
 - από το λογισμικό (π.χ. λειτουργικό σύστημα, database).



Παραλληλισμός I/O-4

Ο παραλληλισμός στοχεύει:

- Κατανομή του φόρτου πολλών μικρών αιτήσεων I/O με στόχο την αύξηση του throughput.
- Παραλληλισμός μεγάλων αιτήσεων I/O με στόχο τη μείωση του χρόνου απόκρισης (response time).

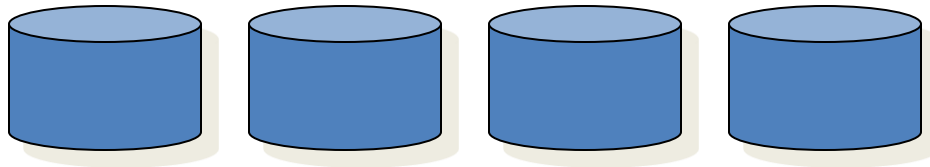
Χρήση RAID

(Redundant Arrays of Inexpensive Disks)



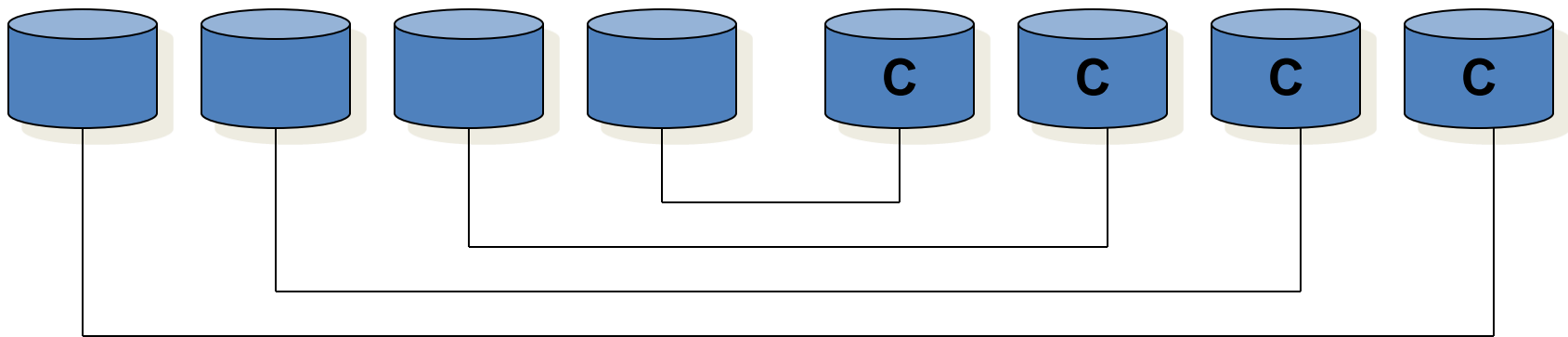
RAID Level 0

- Κάθε δίσκος αποθηκεύει τα δικά του δεδομένα. Δεν υπάρχει επανάληψη.
- Σε περίπτωση που έχουμε failure σε έναν δίσκο τα περιεχόμενα καταστρέφονται.



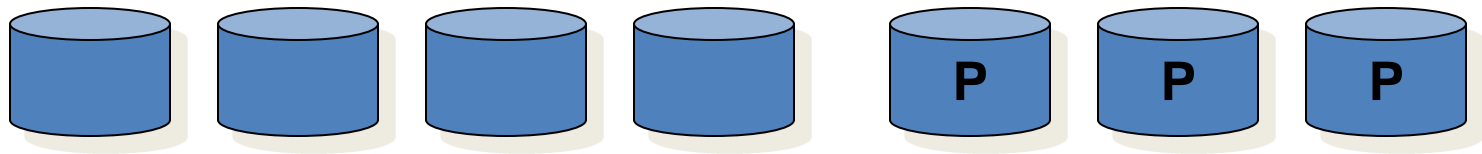
RAID Level 1

- Mirroring
- Κάθε δίσκος έχει και έναν αντίστοιχο ο οποίος περιέχει αντίγραφο των δεδομένων.



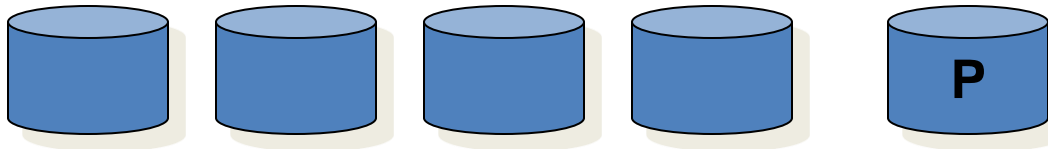
RAID Level 2

- Memory-style error correcting codes
- Για κάθε byte δεδομένων αποθηκεύεται και ένα parity bit, ώστε να μπορεί να γίνει ανακατασκευή δεδομένων σε περίπτωση που έχουμε failure.



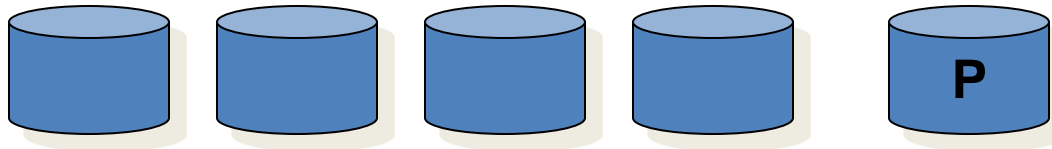
RAID Level 3

- Bit-interleaved parity
- Σε αντίθεση με το level 2 χρησιμοποιεί έναν δίσκο για parity.
- Μειονέκτημα: σε κάθε I/O συμμετέχουν όλοι οι δίσκοι.



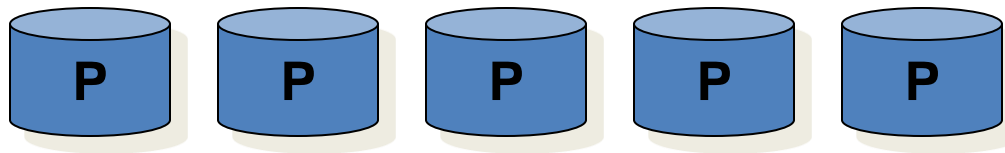
RAID Level 4

- Block-interleaved parity.
- Σε κάθε δίσκο αποθηκεύεται block δεδομένων και χρησιμοποιείται το parity για κάθε block.



RAID Level 5

- Block-interleaved distributed parity.
- Τα parity δεδομένα αποθηκεύονται σε όλους τους δίσκους, μαζί με τα δεδομένα.



Διαμοιρασμός Δεδομένων-1

- Declustering, Partitioning.
- Μέθοδοι διαμοιρασμού των δεδομένων στους διαθέσιμους δίσκους.
- Ο τρόπος διαμοιρασμού των δεδομένων έχει άμεση σχέση με την επίδοση ενός ερωτήματος.
- Στόχοι:
 - Να ενεργοποιούνται όσο το δυνατό λιγότεροι δίσκοι.
 - Οι δίσκοι που ενεργοποιούνται να εκτελούν ισοδύναμες εργασίες, ώστε ο χρόνος επεξεργασίας κάθε δίσκου να είναι περίπου ο ίδιος.



Διαμοιρασμός Δεδομένων-2

Υπάρχουν τρεις διαφορετικές τεχνικές διαμοιρασμού των δεδομένων ενός πίνακα σε ένα σύνολο δίσκων:

- Round-Robin
- Hash Partitioning
- Range Partitioning



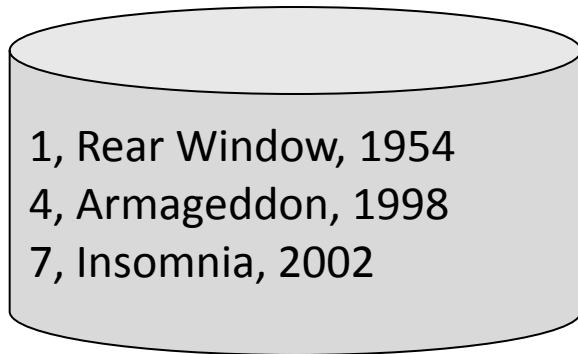
Διαμοιρασμός Δεδομένων-3

ID	Τίτλος Ταινίας	Χρονιά
1	Rear Window	1954
2	Psycho	1960
3	North by Northwest	1959
4	Armageddon	1998
5	Lord of the Rings	2001
6	Minority Report	2002
7	Insomnia	2002

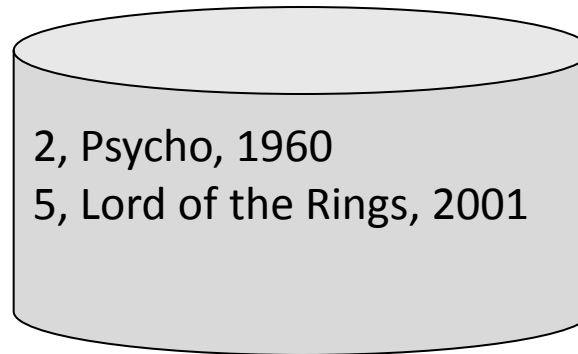


Round-Robin-1

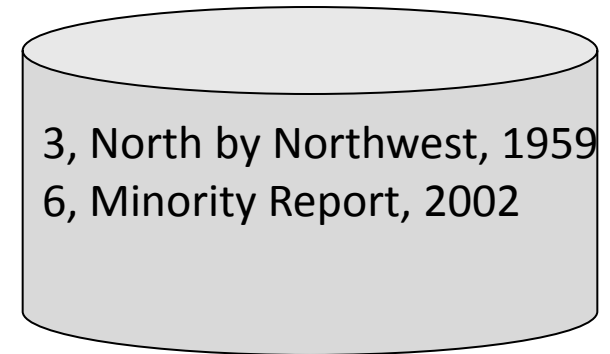
Disk 0



Disk 1



Disk 2



Round-Robin-2

Χρησιμοποιούμε τη συνάρτηση κατακερματισμού
Χρονιά MOD 3

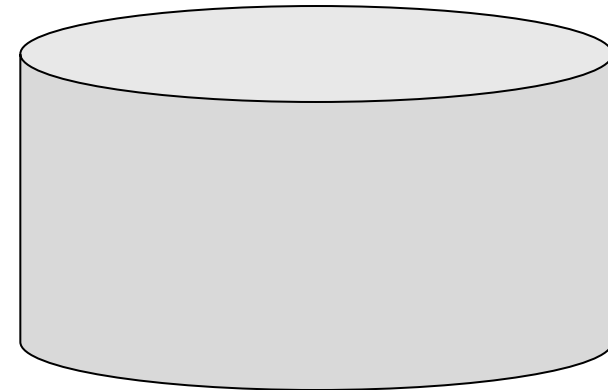
Disk 0

3, North by Northwest, 1959
5, Lord of the Rings, 2001
4, Armageddon, 1998

Disk 1

1, Rear Window, 1954
2, Psycho, 1960
7, Insomnia, 2002
6, Minority Report, 2002

Disk 2



Round-Robin-3

Ο πίνακας χωρίζεται σε περιοχές (ranges) και κάθε περιοχή αποθηκεύεται σε ξεχωριστό δίσκο.

1950 έως 1970

Disk 0

1, Rear Window, 1954
3, North by Northwest, 1959
2, Psycho, 1960

1971 έως 1999

Disk 1

4, Armageddon, 1998

2000 έως 2003

Disk 2

5, Lord of the Rings, 2001
6, Minority Report, 2002
7, Insomnia, 2002



Σύγκριση-4

Ερωτήματα:

SELECT * FROM movies (relation scanning)

SELECT * FROM movies
WHERE title="Armageddon" (point query)

SELECT * FROM movies
WHERE year BETWEEN (1970,2000) (range query)

Να συζητηθεί η απόδοση των μεθόδων διαμοιρασμού για τα ερωτήματα.



Διαμοιρασμός Δομών

- Σε συστήματα πολλών δίσκων υπάρχει η ανάγκη να διαμοιράσουμε τις δομές δεδομένων (π.χ. Β-δένδρα).
- Η έρευνα στο χώρο επικεντρώνεται σε αποδοτικές τεχνικές διαμοιρασμού των σελίδων (blocks) της δομής με στόχο τη γρηγορότερη απάντηση των ερωτημάτων.



Διαμοιρασμός Β-δένδρου-1

Τεχνικές που χρησιμοποιούνται

- Διαμοιρασμός των εγγραφών.
- Υπερ-σελίδες.
- Διαμοιρασμός σελίδων.
- Multi-Disk B-trees.

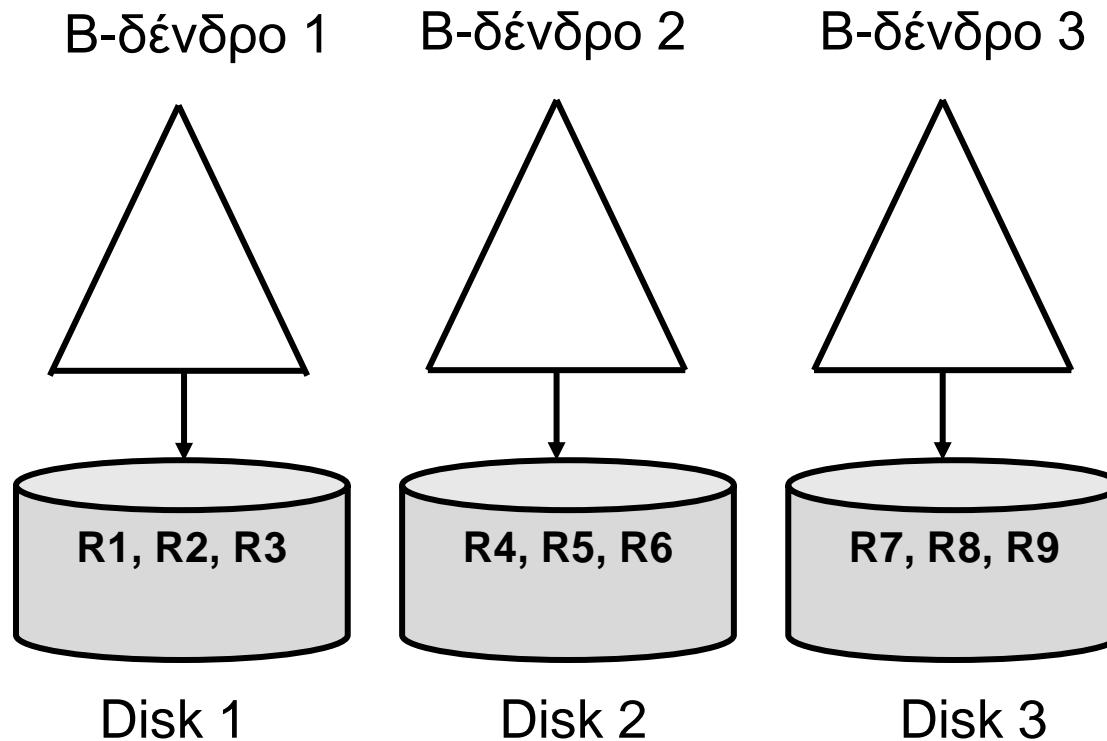


Διαμοιρασμός Εγγραφών-1

- Εφαρμόζεται ένας κανόνας διαμοιρασμού των εγγραφών στους δίσκους (round-robin, hashing, range partitioning).
- Για κάθε δίσκο υπάρχει ένα ξεχωριστό B-δένδρο το οποίο δεικτοδοτεί τις αντίστοιχες εγγραφές.
- Να συζητηθούν τα πλεονεκτήματα και μειονεκτήματα αυτής της προσέγγισης.



Διαμοιρασμός Εγγραφών-2

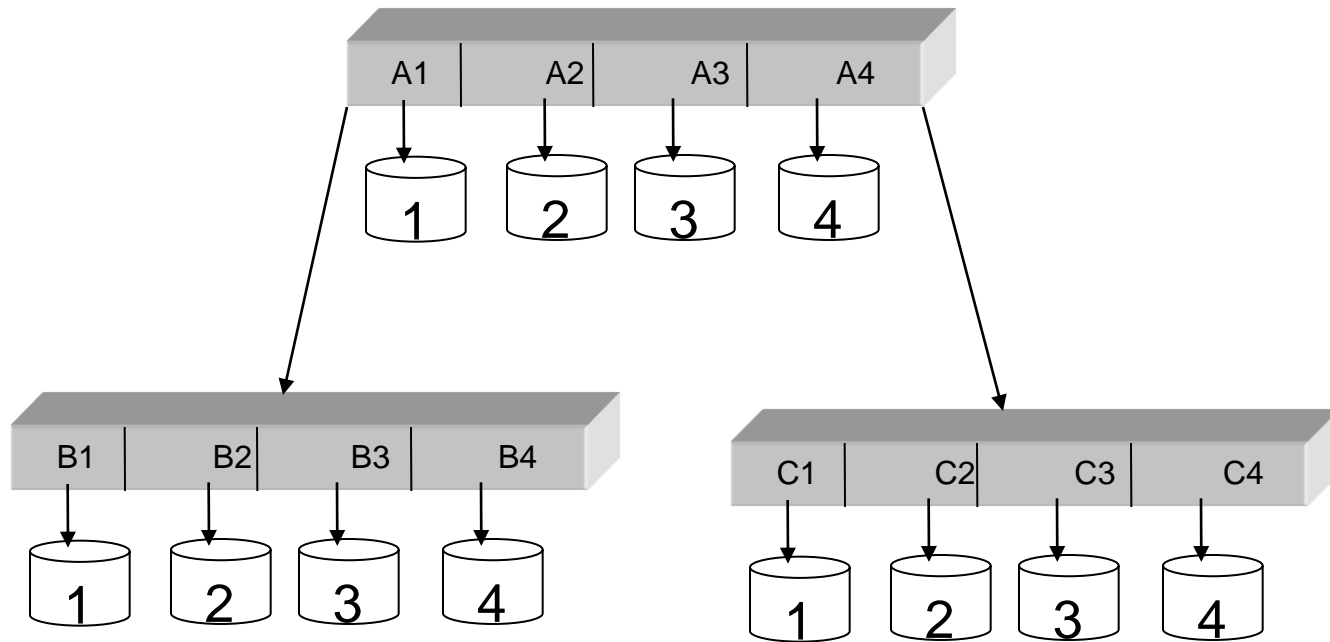


Υπερ-σελίδες-1

- Κάθε κόμβος του δένδρου θεωρείται ότι αποτελείται από N τμήματα, όπου N ο αριθμός των δίσκων. Κάθε τμήμα αποθηκεύεται σε ξεχωριστό δίσκο.



Υπερ-σελίδες-2

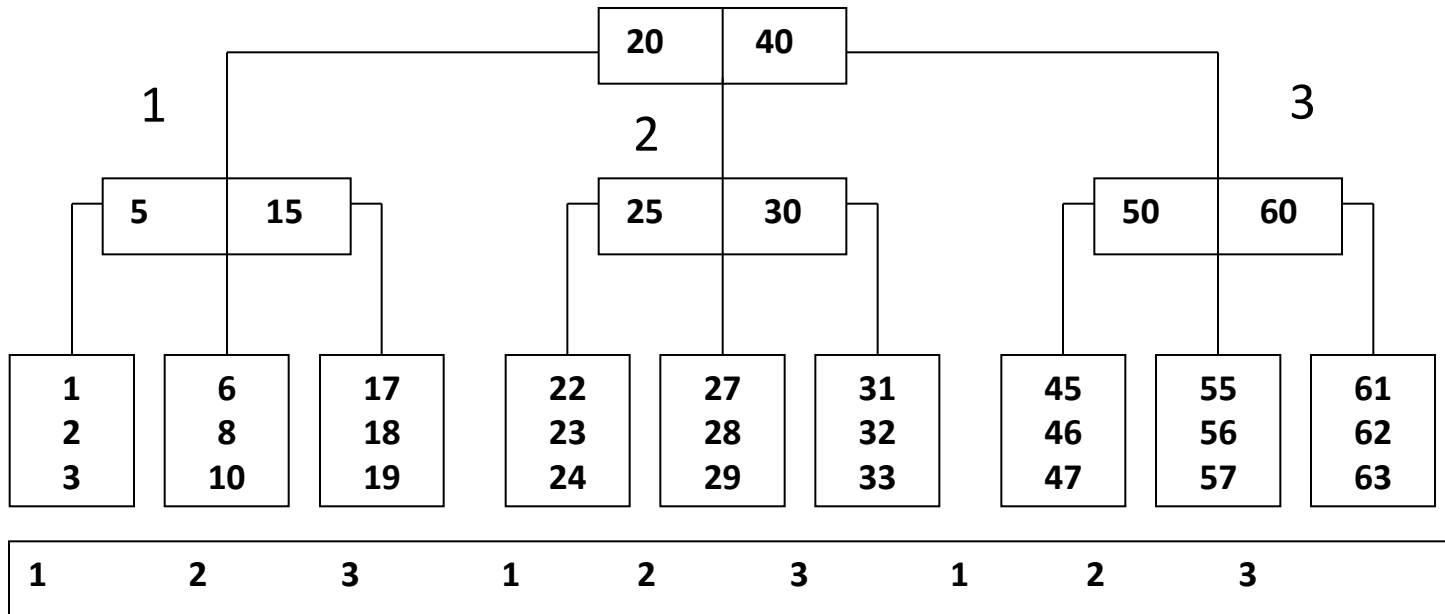


Multi-Disk B-trees-1

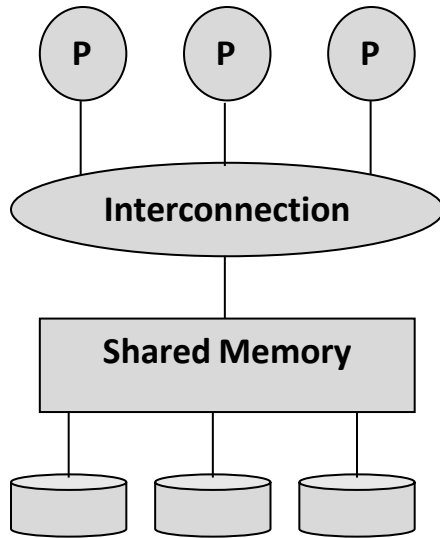
- Οι κόμβοι του δένδρου κατανέμονται στους δίσκους έτσι ώστε, δύο σελίδες που έχουν μεγάλη πιθανότητα να ζητηθούν μαζί σε ένα ερώτημα αποθηκεύονται σε διαφορετικούς δίσκους.
- N αριθμός δίσκων.
- Όταν δημιουργείται μία νέα σελίδα P , αποθηκεύεται στο δίσκο ο οποίος δεν περιέχει τις $N-1$ ή $N-2$ γειτονικές σελίδες με την P .



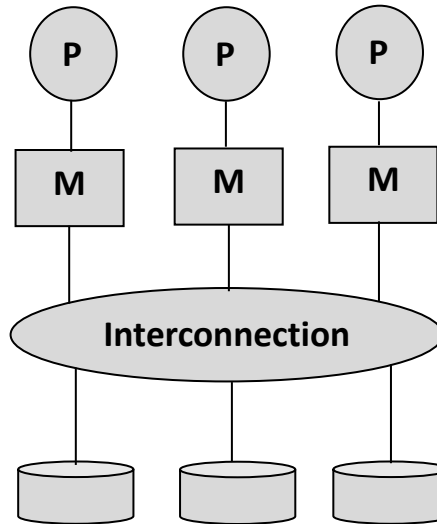
Multi-Disk B-trees-2



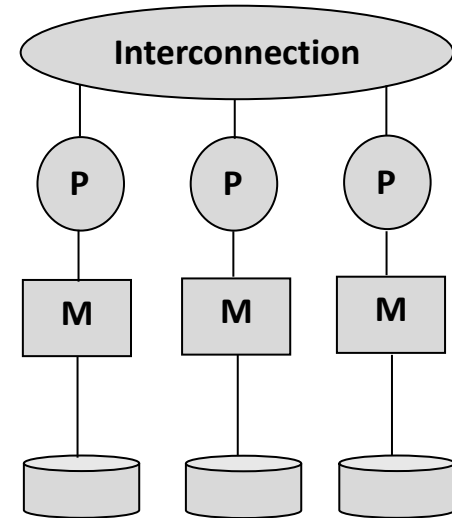
Παράλληλες Βάσεις Δεδομένων



shared everything



shared disk



shared nothing



Μέτρα Επίδοσης

Με ποιον τρόπο μπορούμε να αξιολογήσουμε την επίδοση ενός παράλληλου συστήματος ή ενός παράλληλου αλγορίθμου;

- **Speedup**
- **Sizeup**
- **Scaleup**



Speedup, Scaleup, Sizeup

- **Speedup** = χρόνος σειριακού αλγορίθμου προς το χρόνο παράλληλου αλγορίθμου (θέλουμε γραμμικότητα).
- **Scaleup** = χρόνος επίλυσης μικρού προβλήματος σε μικρό σύστημα προς το χρόνο επίλυσης μεγάλου προβλήματος σε μεγάλο σύστημα (θέλουμε σταθερότητα).
- **Sizeup** = χρόνος επίλυσης μικρού προβλήματος σε μεγάλο σύστημα προς το χρόνο επίλυσης μεγάλου προβλήματος σε μεγάλο σύστημα (θέλουμε γραμμικότητα).



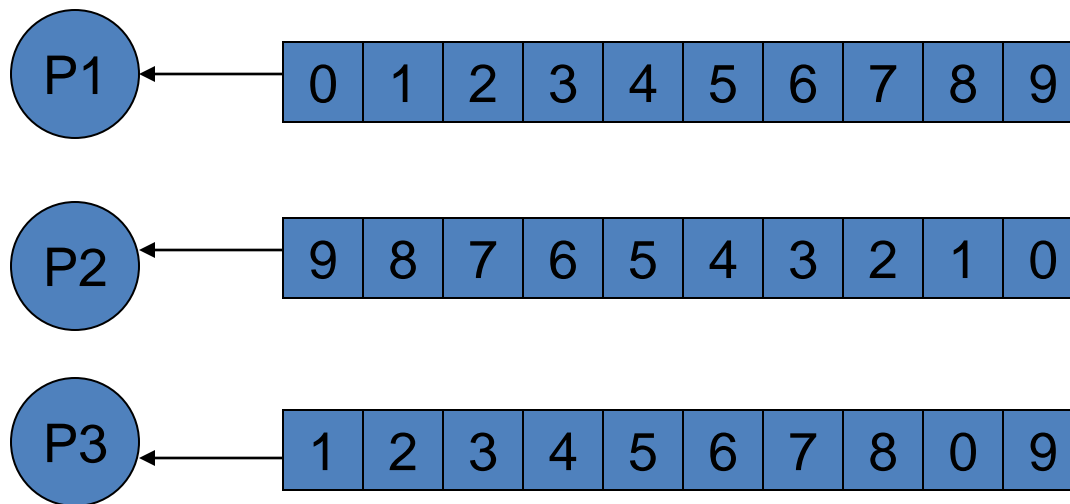
Παράλληλοι Αλγόριθμοι-1

- Βασικοί στόχοι ενός παράλληλου αλγορίθμου είναι:
 - η διάσπαση του αρχικού προβλήματος σε υποπροβλήματα.
 - η αντιστοίχιση υποπροβλημάτων σε επεξεργαστές.
- Ο τρόπος με τον οποίο πραγματοποιούνται τα παραπάνω είναι καθοριστικός για την επίδοση του αλγορίθμου.
- Σε πολλές περιπτώσεις φαίνεται εύκολο, όμως απαιτείται προσοχή στις παγίδες.



Παράλληλοι Αλγόριθμοι-2

Έχουμε 3 arrays με 10 στοιχεία το καθένα, και διαθέτουμε 3 επεξεργαστές.



Παράλληλοι Αλγόριθμοι-3

- Σε πολλές περιπτώσεις απαιτείται να έχουμε μία εκτίμηση σχετικά με το χρόνο που απαιτείται για την επίλυση ενός υποπροβλήματος.
- Η αντιστοίχιση υποπροβλημάτων σε επεξεργαστές μπορεί να μη γίνεται στατικά αλλά δυναμικά, ανάλογα με τις ανάγκες που προκύπτουν κατά την πορεία απάντησης του ερωτήματος (static vs dynamic assignment).



Παράλληλη Ταξινόμηση

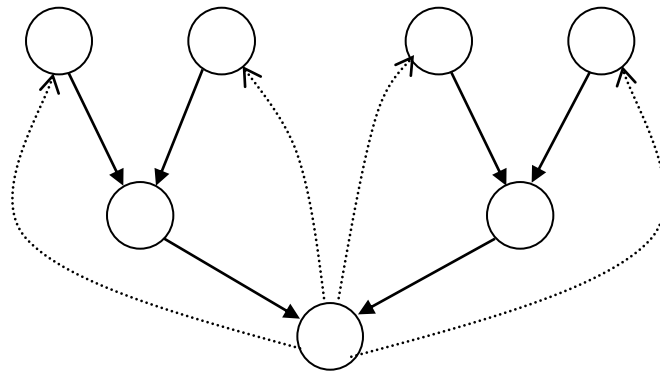
Στη συνέχεια περιγράψουμε δύο παράλληλους αλγορίθμους οι οποίοι λύνουν το πρόβλημα της ταξινόμησης χρησιμοποιώντας N επεξεργαστές.

- Parallel Merge-Sort
- Partitioned-Based Parallel Sorting



Parallel Merge-Sort-1

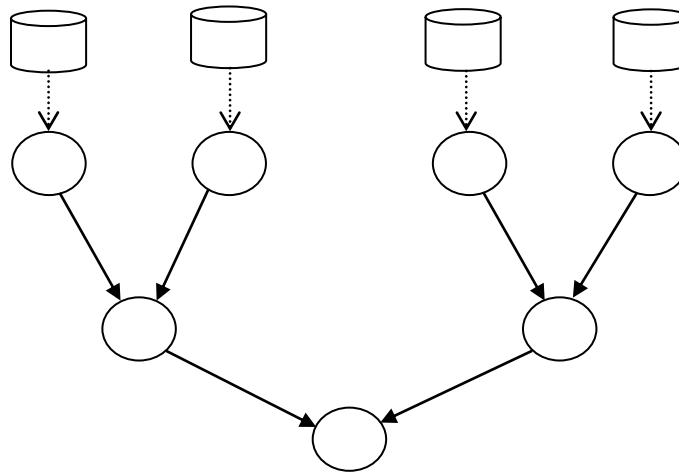
Backend Sort: Το αρχείο προς ταξινόμηση διαμοιράζεται στους επεξεργαστές και μετά την επεξεργασία, το ταξινομημένο αρχείο επιστρέφει στον host.



Parallel Merge-Sort-2

Distributed Sort: αρχικά το αρχείο είναι ήδη διαμοιρασμένο στους επεξεργαστές.

Στο τέλος της ταξινόμησης είτε εγγράφεται στο δίσκο είτε αποστέλλεται στον host.



Parallel Merge-Sort-3

- Ο αλγόριθμος θεωρεί ότι οι επεξεργαστές σχηματίζουν μία δενδρική δομή.
- Κάθε επεξεργαστής που βρίσκεται σε «φύλλο» έχει ένα δίσκο και μπορεί να εκτελέσει ταξινόμηση ανεξάρτητα από τους άλλους.
- Διακρίνουμε δύο φάσεις:
 - Φάση ταξινόμησης (sort phase).
 - Φάση συγχώνευσης (merge phase).



Parallel Merge-Sort-4

- Κατά τη φάση της ταξινόμησης κάθε επεξεργαστής στο επίπεδο των φύλλων ταξινομεί τα δεδομένα που του αντιστοιχούν.
- Κατά τη φάση της συγχώνευσης τα ταξινομημένα τμήματα που έχουν δημιουργηθεί από την προηγούμενη φάση συγχωνεύονται για να δώσουν μεγαλύτερα ταξινομημένα τμήματα.
- Η διαδικασία της συγχώνευσης συνεχίζεται μέχρι να ταξινομηθεί όλο το αρχείο.



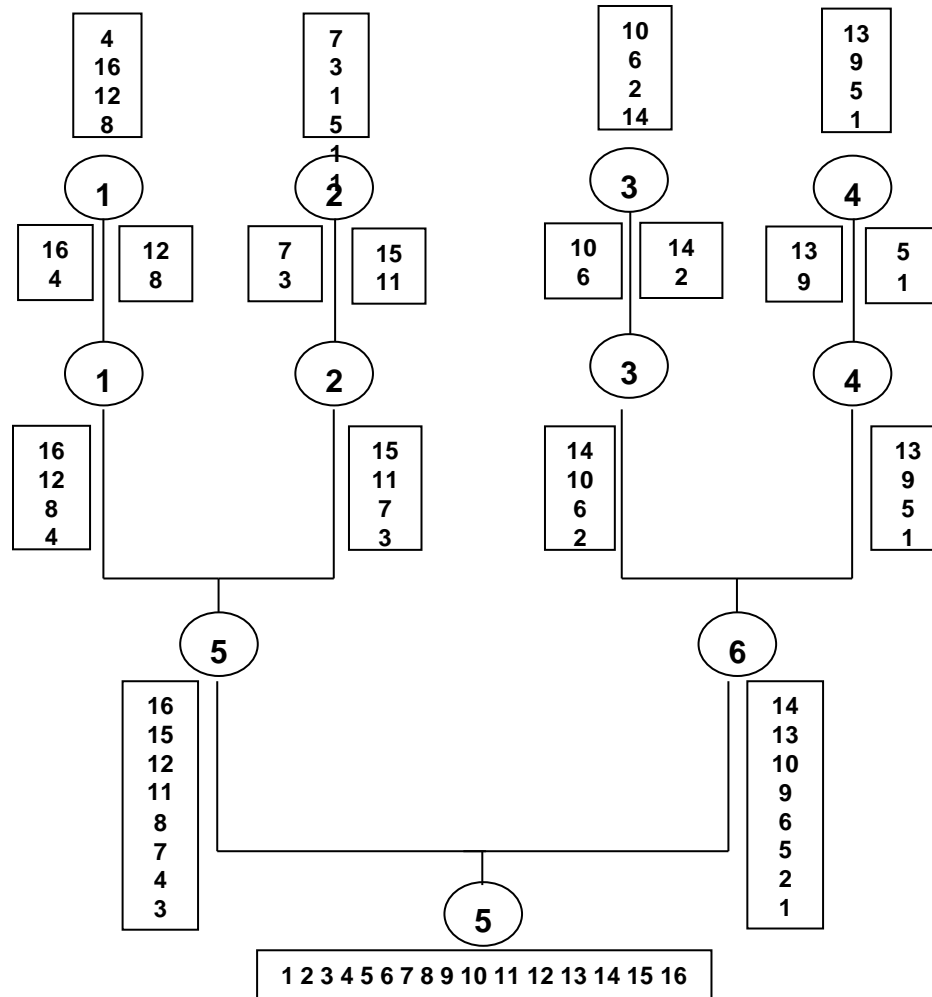
Parallel Merge-Sort-5

Υπάρχουν δύο τρόποι με τους οποίους μπορεί να γίνει η συγχώνευση:

- Με χρήση pipelining μεταξύ των επεξεργαστών διαφορετικών επιπέδων.
- Με χρήση ταυτόχρονης συγχώνευσης στους επεξεργαστές του ίδιου επιπέδου.



Parallel Merge-Sort-6

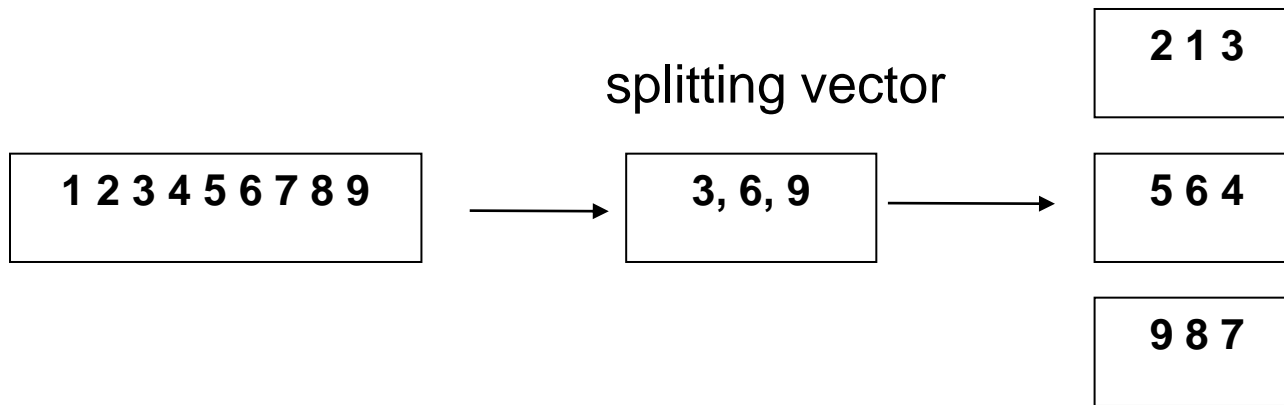


Partitioned-Based Sorting

- Σε κάθε επεξεργαστή αντιστοιχεί ένα εύρος τιμών το οποίο καλείται να ταξινομήσει.
- Η εύρεση του εύρους για κάθε επεξεργαστή καθορίζει και την απόδοση του αλγορίθμου.
- Η μέθοδος προσπαθεί για ομοιόμορφη κατανομή σε όλους τους επεξεργαστές, ώστε να μην υπάρχει bottleneck.



Splitting Vector



Partitioned-Based Sorting-1

- N επεξεργαστές $P_1 \dots P_N$, και κάθε ένας έχει ένα σύνολο δεδομένων D_i .
- Κάθε P_i πραγματοποιεί τυχαία δειγματοληψία στο D_i και αποστέλλει το δείγμα στον συντονιστή (coordinating processor).
- Ο συντονιστής με βάση τα δείγματα καθορίζει ένα splitting vector το οποίο καθορίζει το εύρος τιμών για κάθε επεξεργαστή.
- Το splitting vector αποστέλλεται στους επεξεργαστές και ο κάθε ένας διανέμει τα δεδομένα στους αντίστοιχους επεξεργαστές.
- Κάθε επεξεργαστής ταξινομεί τα δεδομένα του και εγγράφει τα αποτελέσματα στον τοπικό δίσκο.



Partitioned-Based Sorting-2

Προσοχή:

Η δειγματοληψία κοστίζει. Μεγάλο δείγμα σημαίνει καλή προσέγγιση των δεδομένων αλλά μεγάλο κόστος υπολογισμού και μετάδοσης. Μικρό δείγμα σημαίνει μικρό κόστος υπολογισμού και μετάδοσης αλλά υπάρχει ο κίνδυνος για κακή εκτίμηση.



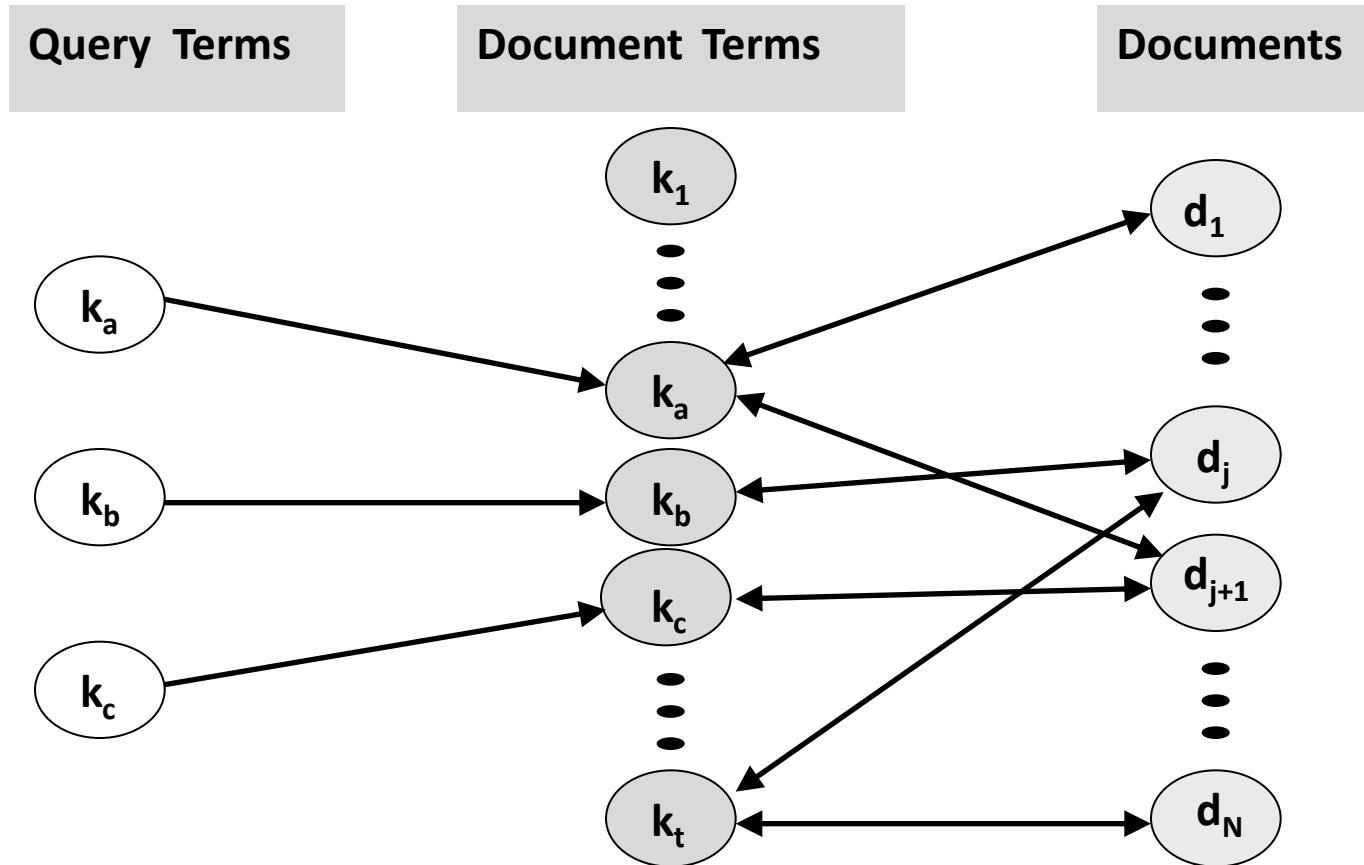
Παράλληλη IR-1

Ο σχεδιασμός παράλληλων αλγορίθμων για IR μπορεί να γίνει με δύο βασικούς τρόπους

- **Δημιουργία νέων τεχνικών** ανάκτησης οι οποίες οδηγούν σε παραλληλισμό.
- **Προσαρμογή των τεχνικών** που ήδη υπάρχουν ώστε να λειτουργήσουν σε ένα παράλληλο περιβάλλον.



Neural Net Model-1



Neural Net Model-2

- Κάθε κόμβος του δικτύου αντιστοιχεί σε έναν επεξεργαστή του συστήματος.
- Ένα νευρωνικό δίκτυο μπορεί να υλοποιηθεί εύκολα σε παράλληλο σύστημα αρχιτεκτονικής SIMD.
- Η χρήση του μοντέλου αυτού για IR μας οδηγεί «από μόνο του» στη χρήση παραλληλισμού. Δεν απαιτούνται μετατροπές για να χρησιμοποιηθεί σε ένα παράλληλο IR σύστημα.



Προσαρμογή Μεθόδων-1

- Έχουμε σαν αφετηρία μία μέθοδο που γνωρίζουμε ότι λειτουργεί σε σειριακό σύστημα (έναν επεξεργαστή).
- Εφαρμόζονται μετατροπές στη μεθοδολογία ώστε να μπορεί να λειτουργήσει σε παράλληλο σύστημα. (Δεν είναι πάντα εύκολο).
- Πρέπει να γνωρίζουμε την παράλληλη αρχιτεκτονική του παράλληλου συστήματος (SIMD, MIMD, κλπ) διότι έχει άμεση σχέση με τον τρόπο λειτουργίας της μεθόδου.



Προσαρμογή Μεθόδων-2

- Έχουν προταθεί αρκετές μέθοδοι οι οποίες λειτουργούν σε παράλληλα συστήματα.
- Μερικές μόνο έχουν παραμείνει και θεωρούνται standards.
- Εστιάζουμε σε αρχιτεκτονικές SIMD και MIMD οι οποίες είναι και οι πιο επικρατέστερες στο χώρο.



Παράλληλη IR-2

- Αρχιτεκτονική MIMD
 - Inverted Files
 - Suffix Arrays
 - Signature Files

- Αρχιτεκτονική SIMD
 - Inverted Files
 - Signature Files



MIMD-1

- Ο πιο απλός τρόπος να εκμεταλλευτούμε ένα MIMD σύστημα είναι με τη χρήση **multitasking**, δηλαδή
- Κάθε επεξεργαστής εκτελεί ένα ξεχωριστό έργο (task) ανεξάρτητα από τους άλλους επεξεργαστές.
- Θεωρούμε ότι κάθε task είναι μία μηχανή αναζήτησης.
- Οι διαφορετικές μηχανές αναζήτησης δε συνεργάζονται μεταξύ τους για τη σύνθεση της απάντησης στο ερώτημα του χρήστη.

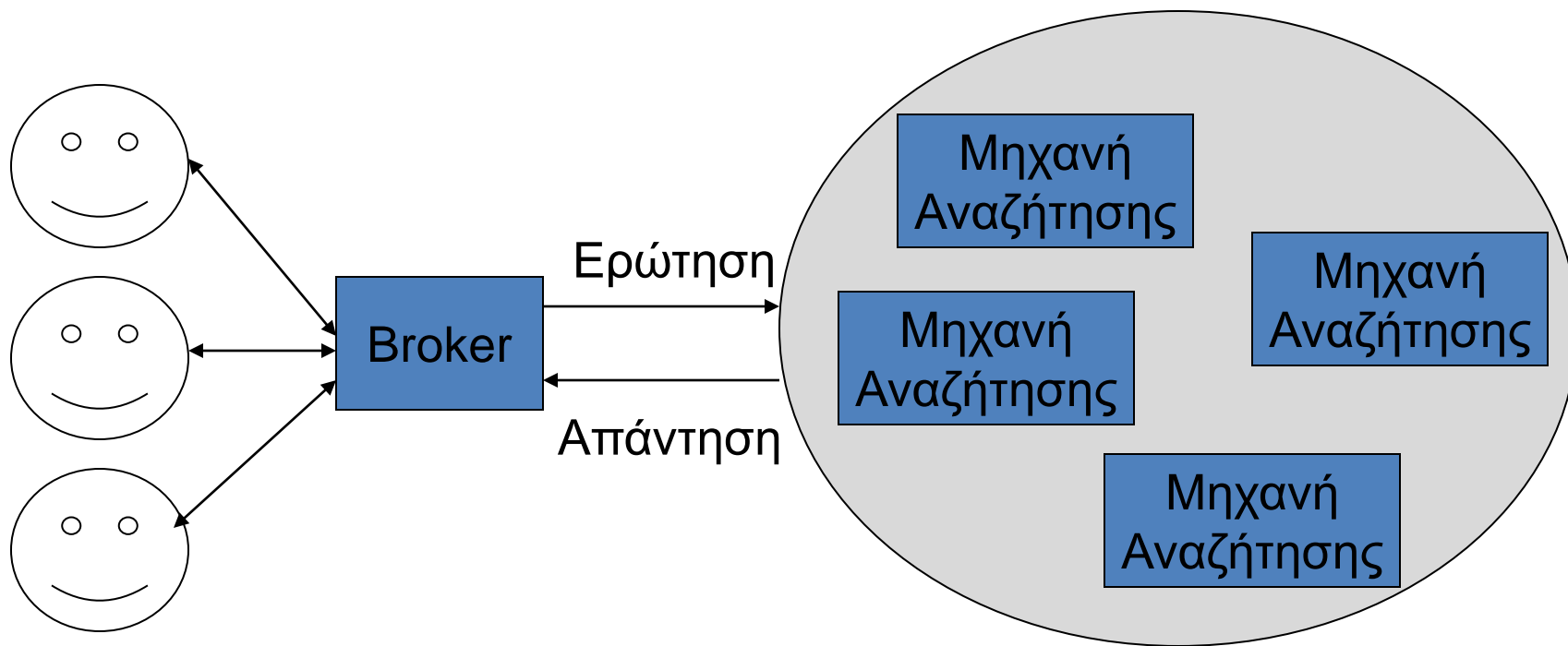


MIMD-2

- Η υποβολή των ερωτήσεων προς το σύστημα πραγματοποιείται με τη βοήθεια ενός μεσολαβητή (broker).
- Ο broker αναλαμβάνει να καταγράψει το ερώτημα του χρήστη και να το τροφοδοτήσει προς τις μηχανές αναζήτησης για επεξεργασία.
- Όσο αυξάνεται ο αριθμός των επεξεργαστών τόσο αυξάνεται και η δυνατότητα του συστήματος να «τρέχει» περισσότερες μηχανές αναζήτησης.



MIMD-3



MIMD-4

- Αν και η μέθοδος είναι αρκετά απλή, απαιτείται προσοχή.
- Όσο αυξάνεται ο αριθμός των επεξεργαστών, αντίστοιχα πρέπει να αυξάνεται και ο αριθμός των δίσκων.
- Οι μηχανές αναζήτησης έχουν απαιτήσεις για I/O. Εάν δημιουργούνται συχνές «συγκρούσεις» μεταξύ αιτήσεων από διαφορετικές μηχανές αναζήτησης, τότε το σύστημα I/O μπορεί να γίνει bottleneck.



MIMD-5

- Σε συνδυασμό με την αύξηση του αριθμού των δίσκων, ο διαχειριστής του συστήματος πρέπει να διαμοιράσει με «έξυπνο» τρόπο τα δεδομένα και τις δομές αναζήτησης στους δίσκους.
- Ένας τρόπος είναι να αποθηκεύσουμε τη δομή αναζήτησης σε κάθε δίσκο.
- Εναλλακτικά μπορούμε να χρησιμοποιήσουμε την τεχνολογία RAID ώστε η λύση να είναι καλύτερη και αποδοτικότερη από άποψη χώρου και ταχύτητας επεξεργασίας.

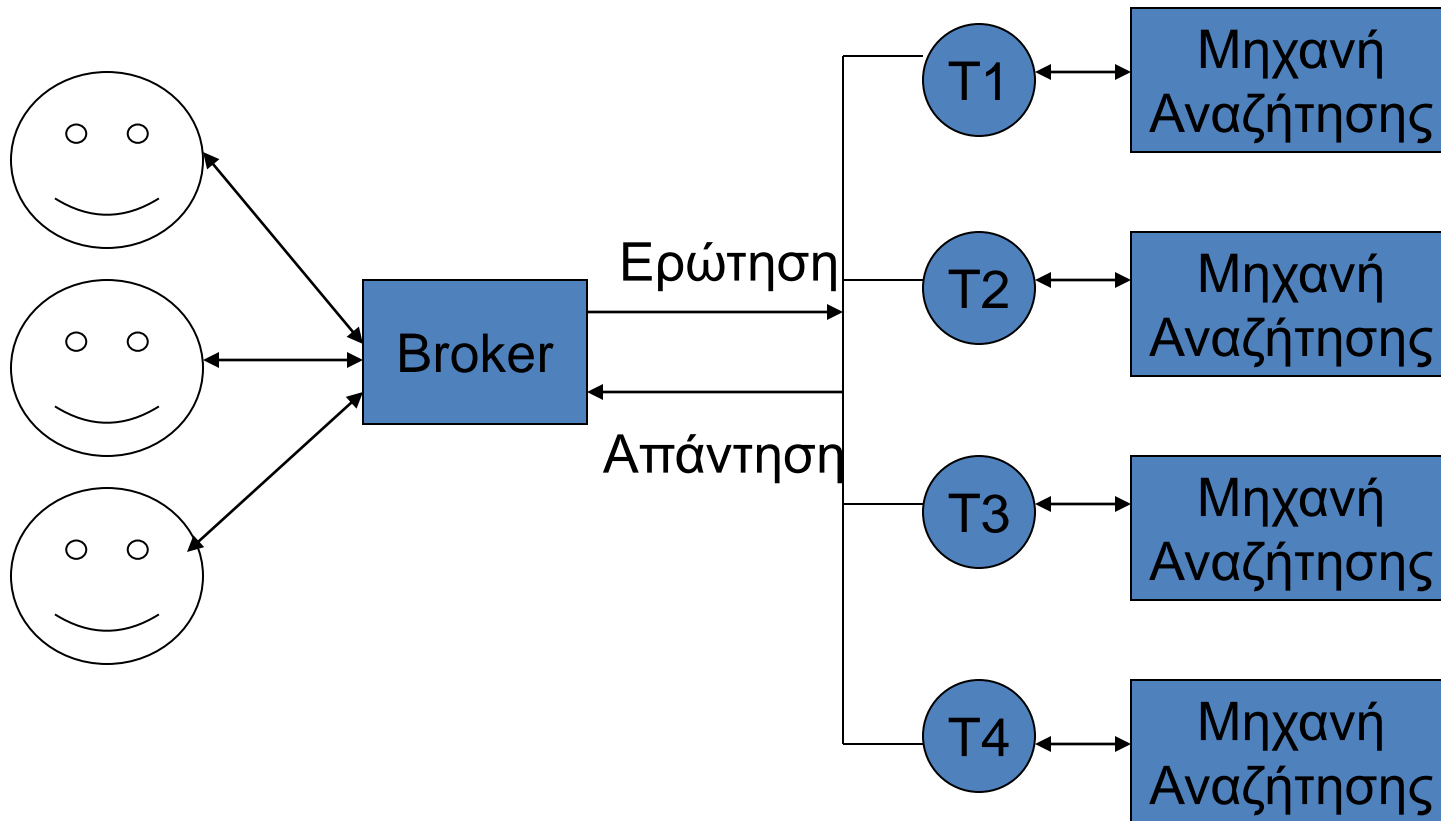


MIMD-6

- Προκειμένου να βελτιώσουμε το σύστημα, μπορούμε να «σπάσουμε» το ερώτημα του χρήστη σε μικρότερα τμήματα, έτσι ώστε κάθε μηχανή αναζήτησης να εκτελέσει ένα μικρό τμήμα του αρχικού ερωτήματος.
- Όταν ο broker δεχθεί ένα ερώτημα, το διασπά σε μικρότερα tasks.
- Κάθε task αποστέλλεται σε μία μηχανή αναζήτησης.
- Τέλος ο broker συλλέγει τα αποτελέσματα από τις μηχανές, συνθέτει την τελική απάντηση και τη μεταβιβάζει στον αντίστοιχο χρήστη.



MIMD-7



Αναπαράσταση Εγγράφων-1

- Αναπαράσταση με index terms (keywords).
- Έχουμε t index terms, k_1, \dots, k_t .
- Έχουμε N έγγραφα D_1, \dots, D_N .
- Σε κάθε ζεύγος (k_i, D_j) αντιστοιχεί ένα βάρος $w_{i,j}$ το οποίο δηλώνει πόσο σημαντικό είναι το keyword i για το έγγραφο j .



Αναπαράσταση Εγγράφων-2

- Τα βάρη $w_{i,j}$ μπορεί να είναι είτε δυαδικά (0 ή 1) είτε πραγματικοί αριθμοί.
- Κάθε έγγραφο D_j μπορεί να αναπαρασταθεί σαν ένα διάνυσμα $\text{vec}(D_j)$ όπου: $\text{vec}(D_j) = (w_{1,j}, \dots, w_{t,j})$.
- Επίσης, κάθε ερώτημα q μπορεί να αναπαρασταθεί σαν ένα διάνυσμα $\text{vec}(q)$ όπου: $\text{vec}(q) = (w_{1,q}, \dots, w_{t,q})$.
- Η μέθοδος IR αναζητά έγγραφα με βάση μία σχέση ομοιότητας
 - $F(\text{vec}(D_j), \text{vec}(q)) = \text{sim}(\text{vec}(D_j), \text{vec}(q))$



Αναπαράσταση Εγγράφων-3

	$k1$	$k2$...	ki	...	kt
$D1$	$w_{1,1}$	$w_{2,1}$...	$w_{i,1}$...	$w_{t,1}$
$D2$	$w_{1,2}$	$w_{2,2}$...	$w_{i,2}$...	$w_{t,2}$
...						
Dj	$w_{1,j}$	$w_{2,j}$...	$w_{i,j}$...	$w_{t,j}$
...						
DN	$w_{1,N}$	$w_{2,N}$...	$w_{i,N}$...	$w_{t,N}$



Διαμοιρασμός-1

Πως θα μοιράσουμε τα δεδομένα; Δύο τρόποι:

- διαμοιρασμός εγγράφων.
- διαμοιρασμός όρων.



Διαμοιρασμός-2

- Ο πίνακας των βαρών τεμαχίζεται με βάση τα έγγραφα (οριζόντια).
- Τα N κείμενα της συλλογής κατανέμονται στους P επεξεργαστές (μηχανές αναζήτησης).
- Έτσι σχηματίζονται P τμήματα και το καθένα περιέχει περίπου N/P έγγραφα.
- Κατά την επεξεργασία του ερωτήματος, κάθε μηχανή αναζήτησης επεξεργάζεται μόνο τα N/P έγγραφα που της αντιστοιχούν.



Διαμοιρασμός Εγγράφων

- Ο πίνακας των βαρών τεμαχίζεται με βάση τα έγγραφα (οριζόντια).
- Τα N κείμενα της συλλογής κατανέμονται στους P επεξεργαστές (μηχανές αναζήτησης).
- Έτσι σχηματίζονται P τμήματα και το καθένα περιέχει περίπου N/P έγγραφα.
- Κατά την επεξεργασία του ερωτήματος, κάθε μηχανή αναζήτησης επεξεργάζεται μόνο τα N/P έγγραφα που της αντιστοιχούν.



Διαμοιρασμός Όρων

- Η μέθοδος τεμαχίζει τον πίνακα βαρών με βάση τα index terms (κάθετα).
- Το σύνολο των index terms χωρίζεται σε P τμήματα.
- Κάθε τμήμα περιέχει ένα υποσύνολο του συνόλου των index terms.
- Επομένως για την αξιολόγηση και βαθμολόγηση ενός κειμένου συμμετέχουν όλοι οι επεξεργαστές.



Μέθοδοι Επεξεργασίας

- Στη συνέχεια αναλύουμε μεθόδους επεξεργασίας που χρησιμοποιούν αντεστραμμένο κατάλογο και στηρίζονται σε:
 - διαμοιρασμό εγγράφων.
 - διαμοιρασμό όρων.



Inverted Files

Διαμοιρασμός εγγράφων:

- Λογικός διαμοιρασμός (logical partitioning).
- Φυσικός διαμοιρασμός (physical partitioning).



Λογικός Διαμοιρασμός-1

- Η κατανομή των εγγράφων στους επεξεργαστές πραγματοποιείται χρησιμοποιώντας την ίδια δομή του αντεστραμμένου αρχείου της σειριακής μεθόδου.
- Το αντεστραμμένο αρχείο επεκτείνεται έτσι ώστε να δώσει σε κάθε μηχανή αναζήτησης δυνατότητα πρόσβασης στο τμήμα του αρχείου το οποίο είναι υπεύθυνο για τη δεικτοδότηση του υποσυνόλου της μηχανής.

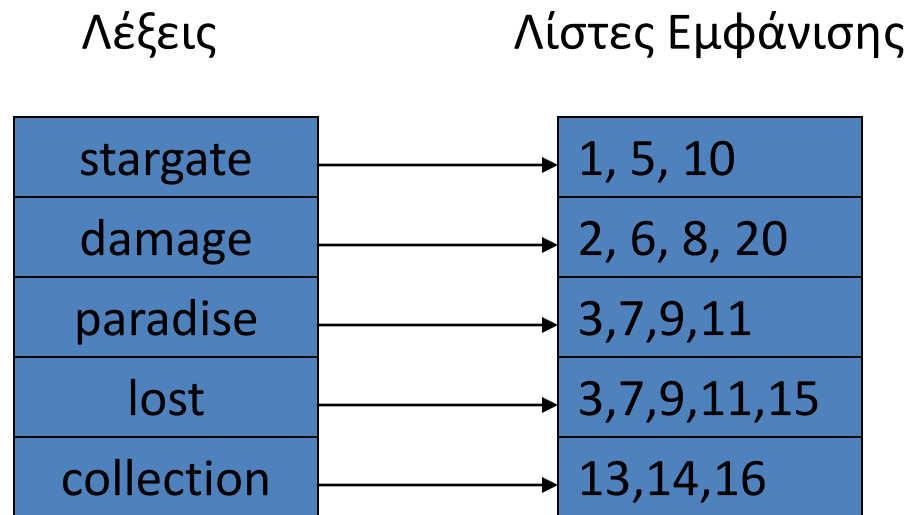


Λογικός Διαμοιρασμός=2

- Κάθε θέση στο λεξικό περιέχει P pointers (P : αριθμός επεξεργαστών).
- Ο j -οστός pointer δεικτοδοτεί τα έγγραφα που βρίσκονται στις λίστες εμφάνισης και πρέπει να επεξεργαστούν από τον επεξεργαστή P_j .



Λογικός Διαμοιρασμός-3

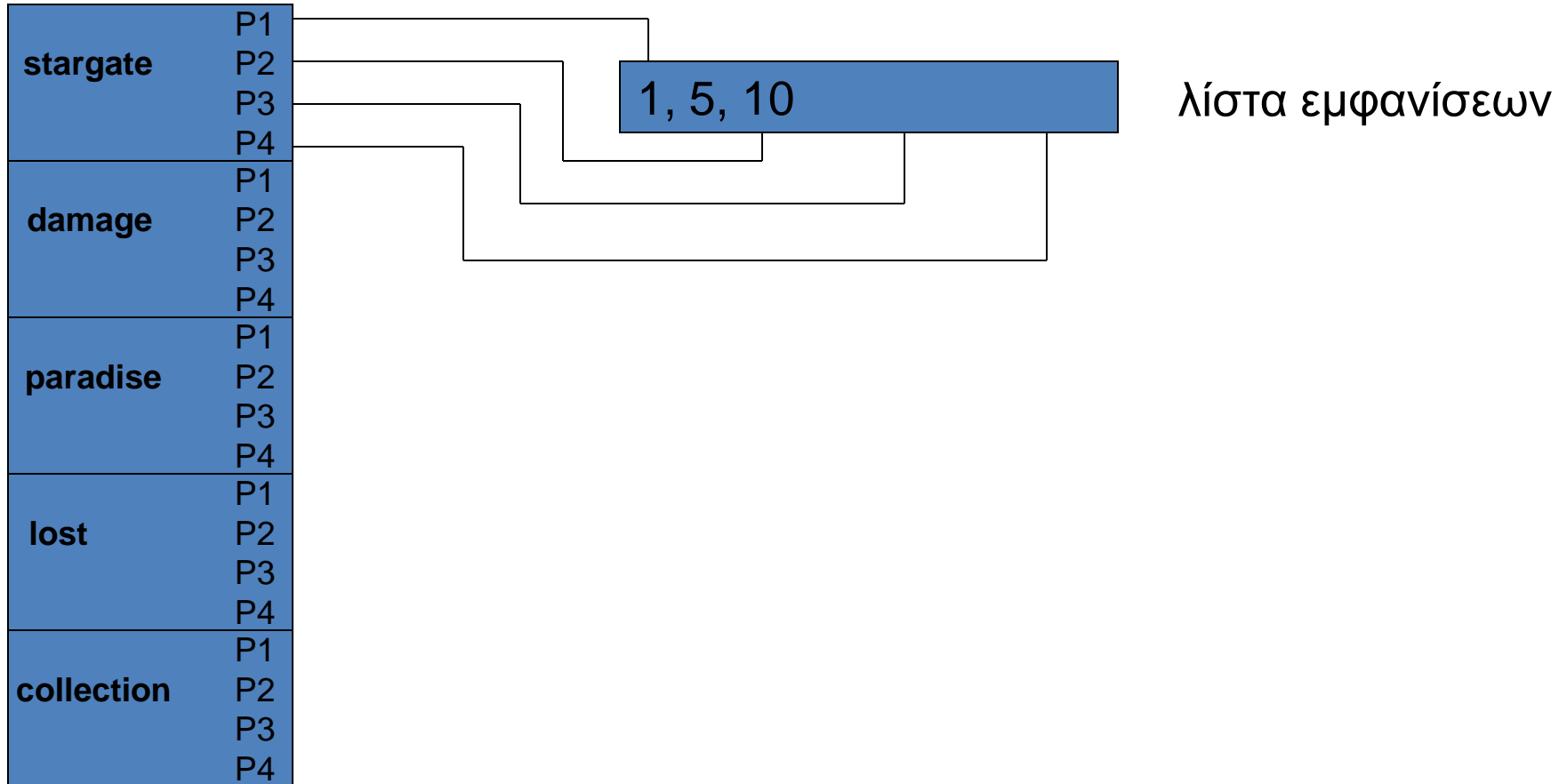


Αντεστραμμένο Αρχείο σε σειριακό περιβάλλον ($P = 1$)



Λογικός Διαμοιρασμός-4

όροι



Λογικός Διαμοιρασμός-5

- Όταν ο broker δέχεται ένα ερώτημα για επεξεργασία, βεβαιώνεται ότι η δομή του αντεστραμμένου αρχείου είναι φορτωμένη στην κοινή μνήμη ώστε να είναι προσπελάσιμη από όλους τους επεξεργαστές.
- Στη συνέχεια ο broker δημιουργεί P παράλληλες διαδικασίες (processes).
- Κάθε διαδικασία εκτελεί τον ίδιο αλγόριθμο βαθμολόγησης στα κείμενα που της αντιστοιχούν.
- Επειδή όλες οι λειτουργίες είναι read-only δεν απαιτείται μηχανισμός κλειδώματος.



Λογικός Διαμοιρασμός-6

- Κάθε διαδικασία επιστρέφει το βαθμό των εγγράφων σε έναν πίνακα βαθμολόγησης ο οποίος είναι κοινός για όλους τους επεξεργαστές.
- Κι εδώ δεν απαιτείται μηχανισμός κλειδώματος διότι κάθε κείμενο βαθμολογείται από μία μόνο διαδικασία.
- Όταν όλοι οι επεξεργαστές ολοκληρώσουν την εργασία τους, ο πίνακας βαθμολόγησης ταξινομείται κατά αύξουσα σειρά και παράγεται η τελική απάντηση στο ερώτημα του χρήστη.



Λογικός Διαμοιρασμός-7

- Κατά τη διάρκεια της κατασκευής του αντεστραμμένου αρχείου μπορούν να χρησιμοποιηθούν **όλοι οι επεξεργαστές** ώστε η κατασκευή να γίνει γρηγορότερα.
- Με τον τρόπο αυτό γίνεται χρήση του παραλληλισμού όχι μόνο κατά τη διαδικασία της αναζήτησης αλλά και κατά την κατασκευή του αντεστραμμένου καταλόγου.

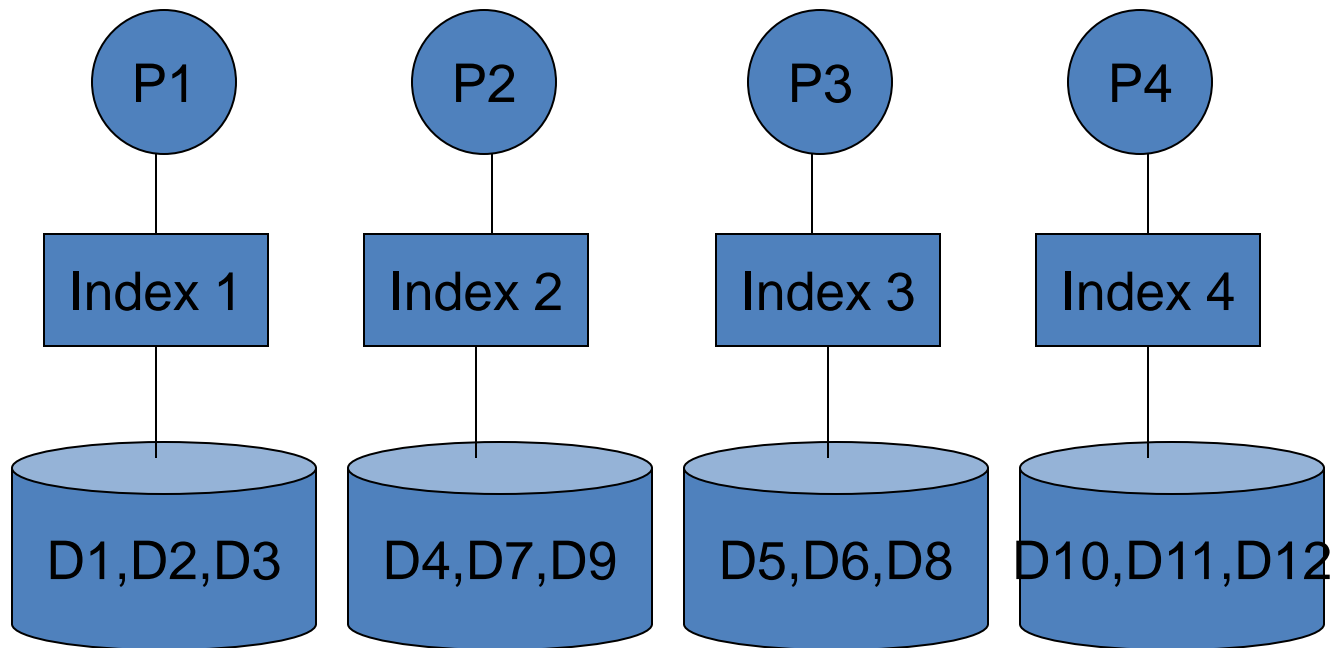


Φυσικός Διαμοιρασμός-8

- Στη μέθοδο αυτή τα έγγραφα κατανέμονται **φυσικά** στους διαθέσιμους επεξεργαστές.
- Σχηματίζονται P συλλογές εγγράφων, μία για κάθε επεξεργαστή.
- Κάθε συλλογή έχει το δικό της αντεστραμμένο κατάλογο. Άρα έχουμε P **ανεξάρτητους** αντεστραμμένους καταλόγους.
- Κάθε επεξεργαστής λειτουργεί με τα δικά του δεδομένα και παράγει έναν ξεχωριστό πίνακα βαθμολόγησης, ταξινομημένο ως προς το βαθμό σχετικότητας μεταξύ ερωτήματος και εγγράφων.



Φυσικός Διαμοιρασμός-9



Φυσικός Διαμοιρασμός-10

- Για να πάρουμε την τελική απάντηση, πρέπει να κατασκευαστεί ο συνολικός πίνακας βαθμολόγησης.
- Γνωρίζουμε ότι ο κάθε πίνακας βαθμολόγησης P_i που παράγεται από κάθε R_i είναι ταξινομημένος κατά αύξουσα σειρά βαθμού.
- Γνωρίζετε κάποιον τρόπο να παραχθεί ο τελικός πίνακας βαθμολόγησης;

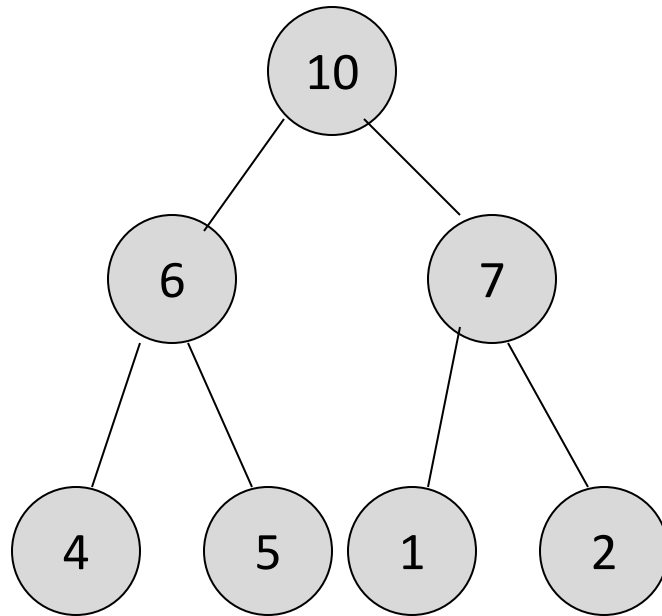


Συγχώνευση Πινάκων-1

- Χρησιμοποιείται **σωρός μεγίστων**.
- Στη δομή αυτή ισχύει ότι το στοιχείο που βρίσκεται στη θέση i είναι μεγαλύτερο από τα στοιχεία που βρίσκονται στις θέσεις $2*i$ και $2*i + 1$.
- Επομένως, το μεγαλύτερο στοιχείο κάθε φορά βρίσκεται στη θέση 1 (πρώτη θέση του πίνακα).



Συγχώνευση Πινάκων-2



1	2	3	4	5	6	7
10	6	7	4	5	1	2

Δενδρική αναπαράσταση πίνακα

Συγχώνευση Πινάκων-3

- Έστω σωρός μεγίστων με n στοιχεία.
- Κόστος εύρεσης του μεγαλύτερου στοιχείου του σωρού μεγίστων $O(1)$.
- Κόστος εισαγωγής ενός στοιχείου $O(\log n)$.
- Κόστος αναδιοργάνωσης σωρού $O(\log n)$.



Συγχώνευση Πινάκων-4

- Δημιουργείται ένας σωρός μεγίστων με P θέσεις (P αριθμός επεξεργαστών).
- Αρχικά παίρνουμε τα πρώτα στοιχεία από τους πίνακες βαθμολόγησης και τα εισάγουμε στο σωρό. Κόστος $O(P * \log P)$.
- Κάθε φορά διαβάζουμε το στοιχείο της ρίζας, το οποίο είναι και το μεγαλύτερο.
- Από τον πίνακα που είχε προέλθει το στοιχείο της ρίζας εισάγουμε το επόμενο στοιχείο στο σωρό.
- Θεωρώντας ότι αναζητούμε τα k έγγραφα με το μεγαλύτερο βαθμό, η διαδικασία αναδιοργάνωσης του σωρού πραγματοποιείται k φορές με συνολικό κόστος $O(k * \log P)$.
- Επομένως, συνολικό κόστος $O((P + k) * \log P)$.



Συγχώνευση Πινάκων-5

Παράδειγμα:

$$P=4$$

$$k=3$$

$$\Pi_1 = [(1,20), (3,15), (5,10), (6,9)]$$

$$\Pi_2 = [(2,12), (4,9), (7,5), (8,3)]$$

$$\Pi_3 = [(9, 30), (10, 25), (11, 5), (12, 1)]$$

$$\Pi_4 = [(14, 4), (13,3), (15,2), (16,1)]$$



Υλοποίηση

- Για τις δύο μεθόδους διαμοιρασμού μπορούμε να χρησιμοποιήσουμε threads ώστε να προγραμματίσουμε τις λειτουργίες του συστήματος IR.
- Κάθε thread εκτελεί τη δική του εργασία.
- Προσφέρεται ένα προγραμματιστικό περιβάλλον ευέλικτο το οποίο επιτρέπει την παράλληλη εκτέλεση εργασιών.
- POSIX threads στις γλώσσες C/C++



Διαμοιρασμός Όρων-1

- Το αντεστραμμένο αρχείο κατασκευάζεται παράλληλα χρησιμοποιώντας τους P επεξεργαστές.
- Οι λίστες εμφάνισης διαμοιράζονται στους επεξεργαστές.
- Για την επεξεργασία ενός ερωτήματος, δημιουργείται το σύνολο των index terms που περιέχονται στο ερώτημα.
- Ο broker αποστέλλει αιτήσεις στους επεξεργαστές ανάλογα με το term που βρίσκεται στο ερώτημα.

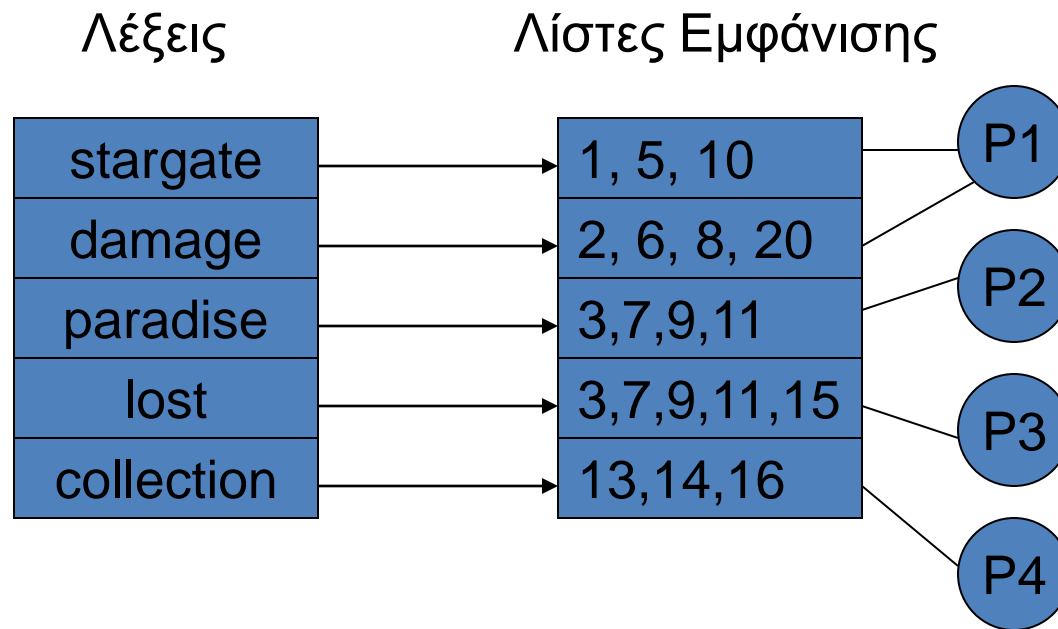


Διαμοιρασμός Όρων-2

- Κάθε επεξεργαστής δημιουργεί έναν «μερικό πίνακα βαθμολόγησης» διότι δεν έχει υπό τον έλεγχό του όλα τα index terms αλλά ένα υποσύνολο αυτών.
- Στο τέλος της επεξεργασίας, οι μερικοί πίνακες βαθμολόγησης συλλέγονται από τον broker ο οποίος παράγει τον τελικό πίνακα ανάλογα με τη συνάρτηση βαθμολόγησης.
- Π..χ. για boolean ερωτήσεις οι μερικοί πίνακες συμμετέχουν στις πράξεις ένωσης, τομής ή αφαίρεσης, ανάλογα με τους λογικούς τελεστές που περιέχονται στο ερώτημα του χρήστη.



Διαμοιρασμός Όρων-3



Σύνοψη

- Η χρήση πολλαπλών πόρων μπορεί να βελτιώσει την απόδοση ενός συστήματος.
- Μελετήσαμε μεθόδους διαμοιρασμού δεδομένων: διαμοιρασμός εγγράφων, διαμοιρασμός όρων.
- Η επεξεργασία του αντεστραμμένου καταλόγου με χρήση πολλών επεξεργαστών βελτιώνει το χρόνο απόκρισης των ερωτημάτων.



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Απόστολος Παπαδόπουλος. «Ανάκτηση πληροφορίας. Παράλληλη Ανάκτηση Πληροφορίας». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://eclass.auth.gr/courses/OCRS388/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>





Τέλος ενότητας

Επεξεργασία: <Μαυρίδης Απόστολος>
Θεσσαλονίκη, <Εαρινό εξάμηνο 2013-2014>





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

