



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΠΛΗΡΟΦΟΡΙΑΚΑ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

**Σχεδιασμός παράλληλων και κατανεμημένων Αλγορίθμων  
για Επεξεργασία Μεγάλων Δεδομένων: Αρχές και κριτήρια  
Σχεδίασης.**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**Μαυρομμάτη Απόστολου**

**Επιβλέπουσα Καθηγήτρια:** Βλάχου Ακριβή, Αναπληρώτρια Καθηγήτρια, Παν/μιο Αιγαίου

**Μέλη εξεταστικής επιτροπής:** Κωστούλας Θεόδωρος, Αναπληρωτής Καθηγητής, Παν/μιο Αιγαίου  
Παπαδοπούλου Πέγκυ, Επίκουρη Καθηγήτρια, Παν/μιο Αιγαίου

Σάμος, Μάιος 2021

## Ευχαριστίες

Η διπλωματική εργασία *Σχεδιασμός παράλληλων και κατανεμημένων Αλγορίθμων για Επεξεργασία Μεγάλων Δεδομένων: Αρχές και κριτήρια Σχεδίασης* εκπονήθηκε την περίοδο Οκτωβρίου 2020 με Μαΐου 2021 στο πλαίσιο ολοκλήρωσης των μεταπτυχιακών μου σπουδών στο πρόγραμμα «Πληροφοριακά και Επικοινωνιακά Συστήματα» του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου.

Με την περάτωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τους καθηγητές/τριες του τμήματος για τις πολύτιμες γνώσεις και συμβουλές που μου προσέφεραν στα γνωστικά αντικείμενα που παρακολούθησα.

Ιδιαίτερα επιθυμώ να ευχαριστήσω την επιβλέπουσα καθηγήτρια κα Βλάχου Ακριβή για την εμπιστοσύνη που μου έδειξε εξ' αρχής στην ανάθεση της εξαιρετικά ενδιαφέρουσας διπλωματικής εργασίας στον τομέα των παράλληλων και κατανεμημένων συστημάτων διαχείρισης δεδομένων, καθώς και κατά τη διάρκεια συγγραφής της. Η καθοδήγησή της, οι πολύτιμες συμβουλές, οι υποδείξεις και η αगाστή συνεργασία μας υπήρξαν πολύτιμοι αρωγοί στην εκπόνηση αυτής της διπλωματικής εργασίας χωρίς τους οποίους θα ήταν αδύνατη η ολοκλήρωσή της.

Τέλος, οφείλω να ευχαριστήσω θερμά την οικογένειά μου και τους γονείς μου για τη συμπαράσταση, την υπομονή και την αγάπη που έλαβα κατά την διάρκεια των μεταπτυχιακών σπουδών και της συγγραφής της παρούσας διπλωματικής εργασίας.

...στα πολυαγαπημένα μου παιδιά Κωνσταντίνο & Πέτρο.

© [2021]

του

Μαυρομάτη Απόστολου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

## Πίνακας περιεχομένων

<b>1</b>	<b>Εισαγωγή</b>	<b>1</b>
1.1	Αρχές και κριτήρια σχεδίασης παράλληλων και κατανεμημένων αλγορίθμων για επεξεργασία μεγάλων δεδομένων	1
1.2	Αντικείμενο της διπλωματικής εργασίας	2
1.3	Στόχοι προς επίτευξη	3
1.4	Δομή της διπλωματικής	4
<b>2</b>	<b>Παράλληλη και κατανεμημένη επεξεργασία δεδομένων</b>	<b>5</b>
2.1	Παράλληλη επεξεργασία δεδομένων	5
2.2	Κατανεμημένη επεξεργασία δεδομένων	6
2.3	Παράλληλο και κατανεμημένο σύστημα διαχείρισης δεδομένων	7
2.4	Εφαρμογές των κατανεμημένων συστημάτων διαχείρισης δεδομένων	10
2.5	Αρχιτεκτονική κατανεμημένων συστημάτων	11
<b>3</b>	<b>Στόχοι της παράλληλης και κατανεμημένης διαχείρισης δεδομένων</b>	<b>13</b>
<b>4</b>	<b>Διαμερισμός και τοποθέτηση των δεδομένων σε κατανεμημένα συστήματα</b>	<b>17</b>
4.1	Κατακερματισμός των δεδομένων (fragmentation)	18
4.2	Αρχιτεκτονικές κατακερματισμού των δεδομένων	19
4.3	Διαμερισμός των δεδομένων με αντίγραφα (replication)	22
<b>5</b>	<b>Κόστος διαχείρισης των κατανεμημένων δεδομένων</b>	<b>25</b>
<b>6</b>	<b>Αρχιτεκτονική μοντέλων παράλληλων και κατανεμημένων συστημάτων</b>	<b>29</b>
6.1	Κατανεμημένα συστήματα	29
6.2	Παράλληλα συστήματα	30
<b>7</b>	<b>Εφαρμογή παράλληλων και κατανεμημένων αλγορίθμων σε περιβάλλον προσομοίωσης</b>	<b>32</b>
7.1	Χρήση αρχιτεκτονικού μοντέλου για τη μελέτη των παράλληλων Αλγορίθμων	32
7.2	Υλοποίηση αλγορίθμων	34
7.3	Ερώτημα 1 <sup>ο</sup> - Επιλογή τιμής	36
7.4	Ερώτημα 2 <sup>ο</sup> - Υπολογισμός πεδίων τιμών	38
7.5	Ερώτημα 3 <sup>ο</sup> - Αναζήτηση καλύτερης τιμής (top K)	42
7.6	Ερώτημα 4 <sup>ο</sup> - Ταξινόμηση αποτελεσμάτων	47
7.7	Συμπεράσματα	50
<b>8</b>	<b>Σύγχρονες εφαρμογές παράλληλης και κατανεμημένης διαχείρισης δεδομένων</b>	<b>52</b>
8.1	Κατηγοριοποίηση των σύγχρονων εφαρμογών	52

8.2	Συμπεράσματα .....	55
<b>Βιβλιογραφία</b> .....		<b>57</b>

## Λίστα Σχημάτων

Σχήμα 1 - Κατανομή των δεδομένων στο δίκτυο υπολογιστών της εμπορικής εταιρείας. ....	35
Σχήμα 2 - Φάση τοπικής επεξεργασίας των δεδομένων (Ερώτημα 1 <sup>ο</sup> ).....	36
Σχήμα 3 - Φάση συγχώνευσης δεδομένων (Ερώτημα 1 <sup>ο</sup> ).....	37
Σχήμα 4 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 2 <sup>ο</sup> , Συνθήκη Α).....	38
Σχήμα 5 - Φάση μεταφοράς δεδομένων (Ερώτημα 2 <sup>ο</sup> , Συνθήκη Α).....	39
Σχήμα 6 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 2 <sup>ο</sup> , Συνθήκη Α).....	39
Σχήμα 7 - Φάση τοπικής επεξεργασίας των δεδομένων (Ερώτημα 2 <sup>ο</sup> , Συνθήκη Β).....	40
Σχήμα 8 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 2 <sup>ο</sup> , Συνθήκη Β).....	41
Σχήμα 9 - Εκτέλεση αλγορίθμου που οδηγεί σε λανθασμένα αποτελέσματα (Ερώτημα 3 <sup>ο</sup> ).....	42
Σχήμα 10 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 3 <sup>ο</sup> , Συνθήκη Α).....	43
Σχήμα 11 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 3 <sup>ο</sup> , Συνθήκη Α).....	44
Σχήμα 12 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 3 <sup>ο</sup> , Συνθήκη Β).....	45
Σχήμα 13 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 3 <sup>ο</sup> , Συνθήκη Β).....	46
Σχήμα 14 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 4 <sup>ο</sup> , Συνθήκη Α).....	47
Σχήμα 15 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 4 <sup>ο</sup> , Συνθήκη Α).....	48
Σχήμα 16 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 4 <sup>ο</sup> , Συνθήκη Β).....	49
Σχήμα 17 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 4 <sup>ο</sup> , Συνθήκη Β).....	50

## Λίστα Πινάκων

Πίνακας 1 – Συγκεντρωτικός πίνακας παρουσίασης κόστους φάσεων δεδομένων βάσει των ερωτημάτων που χρησιμοποιήθηκαν.....	51
Πίνακας 2 – Κόστος ανά φάση επεξεργασίας στις σύγχρονες εφαρμογές.....	54
Πίνακας 3 – Εκτίμηση ορισμού συμμετοχής κάθε φάσης επεξεργασίας βάσει των χαρακτηριστικών του..	56

## Λίστα Εικόνων

Εικόνα 1 - Σειριακή επεξεργασία δεδομένων.....	5
Εικόνα 2 - Παράλληλη επεξεργασία δεδομένων.....	6
Εικόνα 3 - Λειτουργία παράλληλου συστήματος.....	7
Εικόνα 4 - Λειτουργία κατανεμημένου συστήματος.....	7
Εικόνα 5 – Αναπαράσταση τυπικού κατανεμημένου συστήματος (φυσικό επίπεδο).....	11
Εικόνα 6 – Σχεδιασμός αρχιτεκτονικής κατανεμημένου συστήματος (λογικό επίπεδο).....	12
Εικόνα 7 - Παράδειγμα αρχιτεκτονικής οριζοντίου κατακερματισμού.....	20
Εικόνα 8 - Τεχνικές οριζόντιου κατακερματισμού.....	20
Εικόνα 9 - Παράδειγμα αρχιτεκτονικής κάθετου κατακερματισμού .....	21
Εικόνα 10 - Παράδειγμα αρχιτεκτονικής υβριδικού κατακερματισμού με τη μορφή δέντρου κατάτμησης	22
Εικόνα 11 - Σχηματική λειτουργία πλήρη και μερικού διαχωρισμού δεδομένων.....	23
Εικόνα 12 - Λειτουργία επεξεργαστή ερωτημάτων .....	25
Εικόνα 13 - Λειτουργία βελτιστοποιητή ερωτημάτων .....	26
Εικόνα 14 - Γενικά μοντέλα αρχιτεκτονικής παράλληλων και κατανεμημένων συστημάτων .....	30
Εικόνα 15 - Σχηματική απεικόνιση αρχιτεκτονικής παράλληλων συστημάτων (λογικό μοντέλο).....	31
Εικόνα 16 - Αρχιτεκτονική Client /Server (shared-nothing) .....	33



## Ακρωνύμια – Επεξήγηση ορισμών

Mainframes	Υπολογιστές υψηλής απόδοσης
Shared-nothing	Αρχιτεκτονική κατανεμημένου πληροφοριακού συστήματος
Coordinator	Υπολογιστής που έχει τον ρόλο συντονιστή
CPU	Κεντρική Μονάδα Επεξεργασίας
Tightly coupled	Όρος συνεργαζόμενων επεξεργαστών (σφιχτά συνδεδεμένοι)
Chip	Ολοκληρωμένο κύκλωμα
Clusters	Συστοιχία υπολογιστών
VDSL	Very high-bitrate/high speed DSL.Υπηρεσία σύνδεσης στο διαδίκτυο
Fiber	Υπηρεσία σύνδεσης στο διαδίκτυο μέσω οπτικής ίνας
Loosely coupled	Όρος συνεργαζόμενων επεξεργαστών (χαλαρά συνδεδεμένοι)
Servers	Εξυπηρετητής / διακομιστής
Workstations	Σταθμοί εργασίας (η/υ)
WAN	Wide Area Network
LAN	Local Area Network
Hardware	Υλικό μέρος του υπολογιστή
Software	Λογισμικό
Bottleneck	Φαινόμενο συμφόρησης που προκαλεί πρόβλημα στην επεξεργασία δεδομένων
DB2	Σύστημα σχεσιακής βάσης δεδομένων που αναπτύχθηκε από την IBM.
Scalability	Επεκτασιμότητα
Nodes	Κόμβοι
Low-end	Χαμηλών προδιαγραφών
High-end	Υψηλών προδιαγραφών
Internet	Διαδίκτυο
World wide web	Παγκόσμιος ιστός
Peer to Peer	Αρχιτεκτονική κατανεμημένων πληροφοριακών συστημάτων
Napster	Συνδρομητική υπηρεσία διαμερισμού αρχείων μουσικής
Εταιρείες logistics	Εταιρείες που αναλαμβάνουν τη διεκπεραίωση πληθώρα υπηρεσιών για λογαριασμό άλλων επιχειρήσεων ή εταιρειών (π.χ. μεταφορά, συσκευασία, αποθήκευση προϊόντων κ.λπ.)
Cloud	Νέφος (υπολογιστικό)

Middleware	Ενδιάμεσο λογισμικό
Transparent	Διαφανή / αόρατα
Διεπαφή	Επικοινωνία μεταξύ διαφορετικών οντοτήτων (π.χ. λειτουργικών συστημάτων και συσκευών)
Ιεραρχικές δομές	Τα δεδομένα δεν αποθηκεύονται διαδοχικά, αλλά τα στοιχεία συνδέονται ιεραρχικά (π.χ. στοίβα LIFO)
Γραμμικές δομές	Τα δεδομένα οργανώνονται με γραμμική σειρά και συνδέονται το ένα μετά το άλλο (π.χ. δυαδικό δέντρο).
Bit	Binary digit – Βασική μονάδα μέτρησης πληροφορίας
IPv4	Πρωτόκολλο επικοινωνίας διαδικτύου
IPv6	Σύγχρονο πρωτόκολλο επικοινωνίας διαδικτύου (αντικατέστησε το IPv4)
Distributed databases	Καταναμημένες Βάσεις Δεδομένων
Distributed database system / DDBS	Συστήμα διαχείρισης καταναμημένων βάσεων δεδομένων
Administrators	Διαχειριστές
Fragmentation	Κατακερματισμός
Replication	Αναπαραγωγή
Completeness	Πληρότητα
Reconstruction	Συνθήκη ανασύνθεσης
Disjointness	Συνθήκη διαχωρισμού (SQL)
I/O bandwidth	Μέγιστο εύρος μεταφοράς δεδομένων εισόδου/εξόδου
Round-robin	Η απλούστερη τεχνική οριζόντιου διαμερισμού δεδομένων
Hash	Τεχνική οριζόντιου διαμερισμού δεδομένων σύμφωνα με τιμή κάποιου πεδίου
Range	Τεχνική οριζόντιου διαμερισμού δεδομένων βάσει εύρους τιμής πεδίων
Full replication	Πλήρης αντιγραφή δεδομένων
Partial replication	Μερική αντιγραφή δεδομένων
Synchronous replication	Σύγχρονη αντιγραφή δεδομένων
Asynchronous replication	Ασύγχρονη αντιγραφή δεδομένων
Latency	Λανθάνων χρόνος (καθυστέρηση)
Transparency	Διαφάνεια
Queries	Ερωτήματα

SQL	Structure Query Language (Γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων)
Query processor	Επεξεργαστής ερωτημάτων
High level language	Υψηλού επιπέδου γλώσσα (εύκολα κατανοητή από τον άνθρωπο)
Low level language	Χαμηλού επιπέδου γλώσσα (δύσκολα να κατανοηθεί από τον άνθρωπο, καταληπτή από τον υπολογιστή)
Query optimization	Βελτιστοποίηση ερωτημάτων
Query optimizer	Βελτιστοποιητής ερωτημάτων (ειδικό λογισμικό)
Statistics	Στατιστικά
Execution plans	Σχέδια εκτέλεσης
Select	Ερώτημα SQL -Επιλογή
Project	Ερώτημα SQL που επιστρέφει τα χαρακτηριστικά που αναφέρονται σε αυτό
Join	Δυαδική πράξη σύνδεσης πινάκων SQL
Semijoin	Δυαδική πράξη σύνδεσης πινάκων SQL. Είναι ένας τύπος του join.
Access cost	Κόστος πρόσβασης
Secondary storage	Δευτερεύον αποθηκευτικό μέσο
Buffer	Περιοχή της μνήμης που χρησιμοποιείται για την προσωρινή αποθήκευση δεδομένων
Storage cost	Κόστος αποθήκευσης
Computation cost	Κόστος υπολογισμού
Memory uses cost	Κόστος χρήσης της κύριας μνήμης
Communication cost	Κόστος επικοινωνίας (μεταφοράς δεδομένων)
Local processing time	Χρόνος τοπικής επεξεργασίας
CPU cost	Κόστος χρήσης του επεξεργαστή
I/O cost	Κόστος εισόδου/εξόδου των δεδομένων
Total cost	Ολικό κόστος
Unit instruction cost	Κόστος εκτέλεσης μίας εντολής από τον επεξεργαστή
No. of instructions	Αριθμός των εντολών προς εκτέλεση
Unit disk I/O	Χώρος αποθήκευσης σε ένα αποθηκευτικό μέσο
Message initiation	Αρχικοποίηση του μηνύματος που είναι προς εκτέλεση
Transmission	Μετάδοση του μηνύματος
Clients	Πελάτες (αναφέρεται σε πληροφοριακά συστήματα)

Network	Δίκτυο
Shared-memory	Αρχιτεκτονική κατακεντρωμένου πληροφοριακού συστήματος. Κάθε κόμβος διαμοιράζεται της μονάδες μνήμης του δικτύου
Shared-disk	Αρχιτεκτονική κατακεντρωμένου πληροφοριακού συστήματος. Κάθε κόμβος διαμοιράζεται τα αποθηκευτικά μέσα του δικτύου
Local processing	Τοπική επεξεργασία δεδομένων
Merging	Συγχώνευση δεδομένων
Id	Identification (μοναδικός κωδικός ταυτοποίησης)
Communications	Επικοινωνίες, αναφέρεται στη μεταφορά των δεδομένων που πραγματοποιείται κατά τη διάρκεια επικοινωνίας των υπολογιστών
Salary	Μισθός
Sum	Άθροισμα
Department	Υποκατάστημα
Where	Ο όρος χρησιμοποιείται για το φιλτράρισμα των εγγραφών σε ένα ερώτημα SQL
Total	Σύνολο
Top k	Χρησιμοποιείται για τον ορισμό αναζήτησης καλύτερης τιμής σε μία βάση δεδομένων
Max	Μέγιστη τιμή
Group by	Η δήλωση αυτή χρησιμοποιείται για τον καθορισμό ομαδοποίησης (SQL)
Order by	Η δήλωση αυτή χρησιμοποιείται για τον καθορισμό ταξινόμησης (SQL)
Server room	Δωμάτιο διακομιστών
Ethernet	Σύνδεση δικτύου μέσω καλωδίου

## Περίληψη

Ο σχεδιασμός και η χρήση των πρώτων κατανεμημένων πληροφοριακών συστημάτων, κατά τη δεκαετία του '80, υπήρξε μία από τις σπουδαιότερες καινοτομίες στην επιστήμη της πληροφορικής. Μόνο τότε κατέστη πραγματικά εφικτή η αποτελεσματική διαχείριση και εκμετάλλευση μεγάλου όγκου δεδομένων, καθώς η ανακάλυψη όλο και πιο σύγχρονων αυτόνομων πληροφοριακών συστημάτων δεν φαινόταν ικανή να καλύψει τις ανάγκες αποτελεσματικής διαχείρισής τους. Σήμερα, τα κατανεμημένα πληροφοριακά συστήματα χρησιμοποιούνται παντού, σε επιχειρήσεις, οργανισμούς, στο διαδίκτυο, στις τηλεπικοινωνίες, σε εφαρμογές διαχείρισης δεδομένων σε πραγματικό χρόνο, γενικά σε όλες τις περιπτώσεις όπου κεντρικά υπολογιστικά συστήματα είναι αδύνατον να ανταπεξέλθουν στις απαιτήσεις διαχείρισης μεγάλου όγκου δεδομένων. Ο σχεδιασμός και η υλοποίηση της διαχείρισης των κατανεμημένων δεδομένων θεωρείται ένα δύσκολο εγχείρημα, ακόμα και σήμερα, καθώς περιλαμβάνει την παραμετροποίηση πολλών επιμέρους στοιχείων, τα οποία δρουν καταληκτικά στην αποτελεσματική και αποδοτική λειτουργία ενός κατανεμημένου πληροφοριακού συστήματος.

Στην παρούσα διπλωματική εργασία καταγράφονται οι βασικές αρχές σχεδιασμού, καθώς και τα κριτήρια που ορίζουν την αποδοτικότητα της εφαρμογής της παράλληλης και κατανεμημένης επεξεργασίας δεδομένων σε ένα πληροφοριακό σύστημα. Ακολουθώς, πραγματοποιείται η υλοποίηση βασικών αλγορίθμων σε περιβάλλον προσομοίωσης, με σκοπό την αναλυτική παρουσίαση του τρόπου λειτουργίας ενός εικονικού παράλληλου και κατανεμημένου συστήματος διαχείρισης δεδομένων, καθώς και του τρόπου με τον οποίο διαμορφώνεται σε κάθε φάση το κόστος επεξεργασίας των δεδομένων του. Τέλος, σύμφωνα με τις διαπιστώσεις που προκύπτουν από την υλοποίηση των αλγορίθμων στο περιβάλλον προσομοίωσης, κατηγοριοποιούνται οι σύγχρονες εφαρμογές χρήσης παράλληλων και κατανεμημένων συστημάτων διαχείρισης δεδομένων, βάσει των κριτηρίων αποδοτικότητας που θα πρέπει να πληρούν σε κάθε φάση επεξεργασίας των δεδομένων τους.

**Λέξεις Κλειδιά:** *βασικές αρχές, αποδοτικότητα, κατανεμημένη επεξεργασία δεδομένων, αλγόριθμοι, κόστος, σύγχρονες εφαρμογές, κριτήρια.*

## Abstract

The design and use of the first distributed information systems in the 1980s was one of the most important innovations in computer science. It was only then that it became possible to effectively manage and exploit large volumes of data, as the discovery of increasingly modern autonomous information systems did not seem to be able to meet the needs of their effective management. Today, distributed information systems are used everywhere, in companies, organizations, on the internet, in telecommunications, in real-time data management applications, generally in all cases where central computing systems are unable to cope with the demands of large volume data management. The design and implementation of distributed data management is considered a difficult task, even today, as it involves the configuration of many components that ultimately contribute to the efficient and effective operation of a distributed information system.

This dissertation analyses the basic design principles as well as the criteria that define the efficiency of the implementation of parallel and distributed data processing in an information system. Following that, the implementation of basic algorithms in a simulation environment is performed, in order to present in detail, the operation of a virtual parallel and distributed data management system, as well as the way in which its data processing costs are configured in each phase. Finally, according to the findings resulting from the implementation of algorithms in the simulation environment, modern applications for the use of parallel and distributed data management systems are categorized, based on the efficiency criteria that must be met in each phase of their data processing.

**Keywords:** *basic principles, efficiency, distributed data processing, algorithms, costs, modern applications, criteria.*

# 1

## *Εισαγωγή*

### *1.1 Αρχές και κριτήρια σχεδίασης παράλληλων και κατανεμημένων αλγορίθμων για επεξεργασία μεγάλων δεδομένων*

Ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζει η επιστήμη της πληροφορικής, ιδιαίτερα τις τελευταίες δύο δεκαετίες, αποτελεί η διαχείριση του τεράστιου όγκου δεδομένων που δημιουργείται, διακινείται και καταναλώνεται παγκοσμίως στα πληροφοριακά συστήματα. Η γιγάντωση του διαδικτύου, η μεγέθυνση των διαδικτυακών συναλλαγών, οι αυξημένες απαιτήσεις των σύγχρονων επιχειρήσεων για την απόκτηση όλο και περισσότερων δεδομένων, καθώς και η χρήση των μέσων κοινωνικής δικτύωσης από όλο και μεγαλύτερο μέρος του παγκόσμιου πληθυσμού, οδήγησαν την τελευταία 20ετία σε μία έκρηξη της αύξησης των δεδομένων παγκοσμίως. Η αδυναμία των κεντρικών υπολογιστικών συστημάτων (mainframes) να ανταπεξέλθουν αποδοτικά στη διαχείριση και αξιοποίηση μεγάλου όγκου δεδομένων ήταν ήδη φανερή από τη δεκαετία του 80'. Η λύση που προτάθηκε για την αντιμετώπιση του προβλήματος ήταν ο κατακερματισμός και διαμερισμός των πόρων και των δεδομένων, υπολογιστών που συνδέονται σε ένα δίκτυο στοχεύοντας στην επίτευξη αποτελεσματικότερης και αποδοτικότερης διαχείρισης των δεδομένων τους, τα οποία απαιτούσαν ιδιαίτερα μεγάλη υπολογιστική ισχύ. Την ίδια δεκαετία, οι επιστήμονες πειραματίζονται εμπράκτως και στην εφαρμογή των πρώτων κατανεμημένων υπολογιστικών συστημάτων. Η ανάπτυξη και διάδοση του προσωπικού υπολογιστή τα επόμενα χρόνια επέδρασε καταληκτικά στην ευρεία διάδοση τους όπου και διαπιστώθηκε μαζικά, σε επιχειρήσεις και οργανισμούς, η αποτελεσματικότητα στη χρήση τους και η υπεροχή τους σε σχέση με τα κεντρικά υπολογιστικά συστήματα. Επιπρόσθετα, η ανάπτυξη υπολογιστών οι οποίοι αποτελούταν από περισσότερους από έναν επεξεργαστή σε συνδυασμό με την εμφάνιση των πρώτων υπολογιστών με πολυπύρηνους επεξεργαστές, έδωσε επιπλέον ώθηση στη αποτελεσματικότερη λειτουργία των κατανεμημένων συστημάτων, καθώς υπήρχε πλέον η δυνατότητα ταυτόχρονων, παράλληλων υπολογισμών ελαχιστοποιώντας ακόμα περισσότερο την ταχύτητα επεξεργασίας των κατανεμημένων δεδομένων. Σήμερα, όταν αναφερόμαστε σε κατανεμημένα πληροφοριακά συστήματα διαχείρισης δεδομένων είναι αυτονόητο ότι στον όρο συμπεριλαμβάνεται και η παράλληλη επεξεργασία των κατανεμημένων δεδομένων.

Ο όρος παράλληλο και κατανεμημένο υπολογιστικό σύστημα περιλαμβάνει όλους τους τομείς ενός πληροφοριακού συστήματος. Διαθέτει κατανεμημένη υπολογιστική ισχύ, κατανεμημένους πόρους, κατανεμημένα δεδομένα, καθώς και κατανεμημένο λογισμικό τα οποία επικοινωνούν μεταξύ τους προκειμένου να επιτύχουν έναν κοινό σκοπό, την επίλυση απαιτητικών προβλημάτων

τα οποία θα ήταν αν όχι αδύνατον, τουλάχιστον πολύ δύσκολο να επιλυθούν από ένα κεντρικό υπολογιστικό σύστημα. Ένα παράλληλο και καταναμημένο σύστημα διαχειρίζεται μια συλλογή δεδομένων που ενώ σε λογικό επίπεδο ανήκουν στο ίδιο σύστημα, σε φυσικό επίπεδο διαμοιράζονται στους κόμβους ενός δικτύου αλληλοσυνδεόμενων υπολογιστών. Πρωτεύοντα ρόλο στην αποδοτικότητα ενός παράλληλου και καταναμημένου συστήματος παίζει ο σχεδιασμός των αλγορίθμων, οι οποίοι έχουν ως βασικό σκοπό τη διαχείριση των πόρων και των δεδομένων του συστήματος με όσο το δυνατόν πιο αποδοτικό τρόπο, δηλαδή με το μικρότερο δυνατό κόστος, τόσο όσον αφορά στην κατανάλωση των πόρων του συστήματος όσο και στην ταχύτητα διεκπεραίωσης των ερωτημάτων.

## 1.2 Αντικείμενο της διπλωματικής εργασίας

Στην παρούσα διπλωματική θα μελετηθούν και θα καταγραφούν οι βασικές αρχές της παράλληλης και καταναμημένης επεξεργασίας μεγάλων δεδομένων. Ένα παράλληλο και καταναμημένο σύστημα αποτελείται συνήθως από ένα πολύπλοκο πληροφοριακό σύστημα που βασίζει τη λειτουργία του στη συνεργασία μιας ομάδας αλληλοσυνδεόμενων υπολογιστών, οι οποίοι με τη χρήση κατάλληλου λογισμικού, έχουν ως βασικό σκοπό την αποδοτική διαχείριση των δεδομένων τους. Θεμελιώδης παράγοντας που ορίζει σε μεγάλο βαθμό την αποδοτικότητα του συστήματος αποτελεί ο τρόπος σχεδιασμού και υλοποίησης των αλγορίθμων που είναι υπεύθυνοι για τη διαχείριση των δεδομένων αυτών. Ο σχεδιασμός και η εφαρμογή των αλγορίθμων αυτών θα πρέπει να βασίζεται σε συγκεκριμένες αρχές και κανόνες διαχείρισης παράλληλων και καταναμημένων δεδομένων προκειμένου να επιτευχθεί η προσδοκώμενη αποδοτικότητα.

Για να γίνει κατανοητή η λειτουργία και εφαρμογή των παράλληλων και καταναμημένων αλγορίθμων θα δημιουργηθεί ένα εικονικό περιβάλλον προσομοίωσης ενός καταναμημένου πληροφοριακού συστήματος το οποίο θα αποτελείται: **α)** από ένα καταναμημένο πληροφοριακό σύστημα, αρχιτεκτονικής shared-nothing, το οποίο διαχειρίζεται ένα σύνολο δεδομένων, **β)** από ένα σύνολο δεδομένων το οποίο περιλαμβάνει έναν διδιάστατο πίνακα δεδομένων ο οποίος έχει κατακερματιστεί ισομερώς στους διακομιστές (*servers*) του συστήματος έχοντας τον ρόλο μίας καταναμημένης βάσης δεδομένων.

Στο εικονικό αυτό περιβάλλον θα τεθούν τέσσερα βασικά, ευρέως χρησιμοποιούμενα, ερωτήματα επίλυσης προβλημάτων. Για κάθε ένα ερώτημα που καλείται να απαντήσει το σύστημα θα παρουσιαστούν οι φάσεις επεξεργασίας των δεδομένων καθώς και το εκτιμώμενο κόστος κάθε φάσης, ξεκινώντας από την στιγμή που ο συντονιστής (*coordinator server*) του συστήματος λάβει το ερώτημα από τον χρήστη, έως και την τελική απάντηση στον χρήστη. Το κόστος κάθε ενέργειας κατά τα στάδια επεξεργασίας των δεδομένων, το οποίο λειτουργεί προσθετικά για τον υπολογισμό του τελικού, συνολικού κόστους, είναι που καθορίζει την αποτελεσματικότητα των αλγορίθμων σε ένα παράλληλο και καταναμημένο πληροφοριακό σύστημα. Στα πιο σύνθετα ερωτήματα που καλείται το σύστημα να απαντήσει θα εφαρμόζεται και εναλλακτικός τρόπος λειτουργίας των αλγορίθμων παρουσιάζοντας τους όρους και τις συνθήκες που επιτρέπουν τις διαφοροποιήσεις,



τόσο όσο αφορά στο κόστος κάθε φάσης επεξεργασίας, όσο και στο συνολικό κόστος ολοκλήρωσης του ερωτήματος.

Ο τρόπος με τον οποίο οι αλγόριθμοι αυτοί διεκπεραιώνουν τα ερωτήματα δεν σχετίζεται αποκλειστικά με το συγκεκριμένο εικονικό πληροφοριακό σύστημα που θα χρησιμοποιηθεί. Σκοπός της παρουσίασης του γενικού αυτού μοντέλου είναι να λειτουργήσει ως οδηγός για την κατανόηση του τρόπου λειτουργίας των αλγορίθμων στη διαχείριση των παράλληλων και κατανεμημένων δεδομένων. Μελετώντας τον τρόπο λειτουργίας τους είναι δυνατόν να εκμαιευτούν τα κριτήρια εφαρμογής των αλγορίθμων για την επίτευξη αποδοτικότητας και σε λοιπά σύγχρονα κατανεμημένα συστήματα τα οποία διαφοροποιούνται. Η διαφοροποίηση αυτή μπορεί να αφορά τις συνθήκες, τους συντελεστές και την πλατφόρμα υλοποίησης κάθε συστήματος. Χωρίζοντας, λοιπόν, σύμφωνα με τα παραπάνω, τις σύγχρονες εφαρμογές των παράλληλων και κατανεμημένων συστημάτων σε κατηγορίες, θα παρουσιαστούν τα κριτήρια που θα πρέπει να πληρούν οι αλγόριθμοι έτσι ώστε κάθε σύστημα να λειτουργεί αποδοτικά, σύμφωνα με τις εκτιμήσεις κόστους της κάθε φάσης επεξεργασίας των δεδομένων τους.

### **1.3 Στόχοι προς επίτευξη**

Σκοπό της παρούσας διπλωματικής αποτελεί η μελέτη και καταγραφή των βασικών αρχών και κριτηρίων σχεδίασης των παράλληλων και κατανεμημένων αλγορίθμων επεξεργασίας μεγάλων δεδομένων. Για την επίτευξη του σκοπού αυτού θα πρέπει αρχικά να παρουσιαστεί το γενικό πλαίσιο λειτουργίας των παράλληλων και κατανεμημένων συστημάτων διαχείρισης δεδομένων. Το πλαίσιο αυτό περιλαμβάνει τις εφαρμογές, τους στόχους, την κατανομή των δεδομένων, το κόστος λειτουργίας καθώς και τις διαθέσιμες αρχιτεκτονικές των παράλληλων και κατανεμημένων συστημάτων διαχείρισης δεδομένων. Ακόλουθο ζητούμενο αποτελεί η εφαρμογή βασικών αλγορίθμων σε περιβάλλον προσομοίωσης. Θα δημιουργηθεί ένα εικονικό περιβάλλον διαχείρισης παράλληλων και κατανεμημένων δεδομένων με σκοπό να εξεταστεί η συμπεριφορά των αλγορίθμων κατά την εκτέλεση και ολοκλήρωση βασικών ερωτημάτων. Για κάθε ερώτημα που καλείται το σύστημα να απαντήσει θα παρουσιάζονται τα στάδια κάθε φάσης επεξεργασίας των δεδομένων που πραγματοποιεί το σύστημα, από την αρχική υποβολή του ερωτήματος, έως ότου το τελικό αποτέλεσμα φτάσει στο χρήστη. Στόχο της υλοποίησης του παραδείγματος αποτελεί η μελέτη της συμπεριφοράς του συστήματος κατά την διάρκεια εκτέλεσης των ερωτημάτων καθώς και της εκτίμησης του κόστους κάθε φάσης επεξεργασίας των δεδομένων του κατανεμημένου συστήματος προσομοίωσης. Τα συμπεράσματα που θα προκύψουν από την εφαρμογή των αλγορίθμων του παραδείγματος θα αποτελέσουν τη βάση για την κατηγοριοποίηση των σύγχρονων εφαρμογών της παράλληλης και κατανεμημένης επεξεργασίας δεδομένων, καθώς και των κριτηρίων που θα πρέπει να πληρούν για την επίτευξη αποδοτικότητας.

## **1.4 Δομή της διπλωματικής**

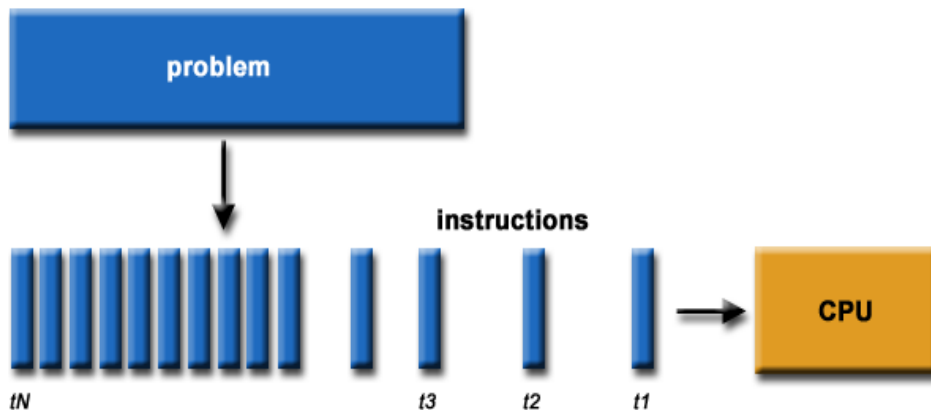
Στην εισαγωγή (κεφάλαιο 1<sup>ο</sup>) γίνεται μία συνοπτική περιγραφή της παράλληλης και κατανεμημένης επεξεργασίας δεδομένων καθώς και των συνθηκών που ευνόησαν τη δημιουργία και εξέλιξη των παράλληλων και κατανεμημένων πληροφοριακών συστημάτων. Επιπλέον, ορίζεται ο σκοπός εκπόνησης της διπλωματικής εργασίας και παρουσιάζονται συνοπτικά η μέθοδος και οι διεργασίες ολοκλήρωσης των στόχων της. Το 2<sup>ο</sup> κεφάλαιο μας εισάγει στις βασικές έννοιες της παράλληλης και κατανεμημένης επεξεργασίας δεδομένων, των παράλληλων και κατανεμημένων συστημάτων διαχείρισης δεδομένων, ενώ στη συνέχεια γίνεται μία σύντομη αναφορά τις σύγχρονες εφαρμογές τους. Στο 3<sup>ο</sup> κεφάλαιο περιγράφονται αναλυτικά οι βασικοί στόχοι της παράλληλης και κατανεμημένης διαχείρισης δεδομένων. Στο 4<sup>ο</sup> κεφάλαιο αναλύεται ο τρόπος και οι πρακτικές με τις οποίες τα δεδομένα, συγκροτημένα ως μια κατανεμημένη βάση δεδομένων, που περιλαμβάνονται σε ένα κατανεμημένο πληροφοριακό σύστημα, κατακερματίζονται και διανέμονται σε ένα αλληλοσυνδεδεμένο δίκτυο υπολογιστών. Στο 5<sup>ο</sup> κεφάλαιο πραγματοποιείται μια αναλυτική περιγραφή του κόστους, του βασικότερου κριτηρίου αποδοτικότητας ενός παράλληλου και κατανεμημένου αλγορίθμου. Στο 6<sup>ο</sup> κεφάλαιο παρουσιάζονται οι βασικές μορφές αρχιτεκτονικής, οι οποίες καθορίζουν τη δομή ενός παράλληλου και κατανεμημένου πληροφοριακού συστήματος. Σε αυτό το κεφάλαιο ολοκληρώνεται η θεωρητική προσέγγιση που αφορά στις αρχές και τα κριτήρια σχεδίασης των παράλληλων και κατανεμημένων αλγορίθμων για επεξεργασία μεγάλων δεδομένων. Στο 7<sup>ο</sup> κεφάλαιο υλοποιείται ένα εικονικό πληροφοριακό σύστημα κατανεμημένων δεδομένων. Η υλοποίηση βασικών αλγορίθμων σε περιβάλλον προσομοίωσης θα μας επιτρέψει να κατανοήσουμε τη λειτουργία αλλά και να εντοπίσουμε τους παράγοντες που διαμορφώνουν το κόστος σε κάθε φάση επεξεργασίας των δεδομένων. Στο 8<sup>ο</sup> κεφάλαιο καταγράφονται και κατηγοριοποιούνται τα σημαντικότερα κριτήρια που ορίζουν την αποδοτικότητα των σύγχρονων εφαρμογών παράλληλων και κατανεμημένων πληροφοριακών συστημάτων. Η προσέγγιση αυτή πραγματοποιείται βάσει των παραγόντων που εντοπίσαμε στο παράδειγμα του προηγούμενου κεφαλαίου.

# 2

## Παράλληλη και καταναμημένη επεξεργασία δεδομένων

### 2.1 Παράλληλη επεξεργασία δεδομένων

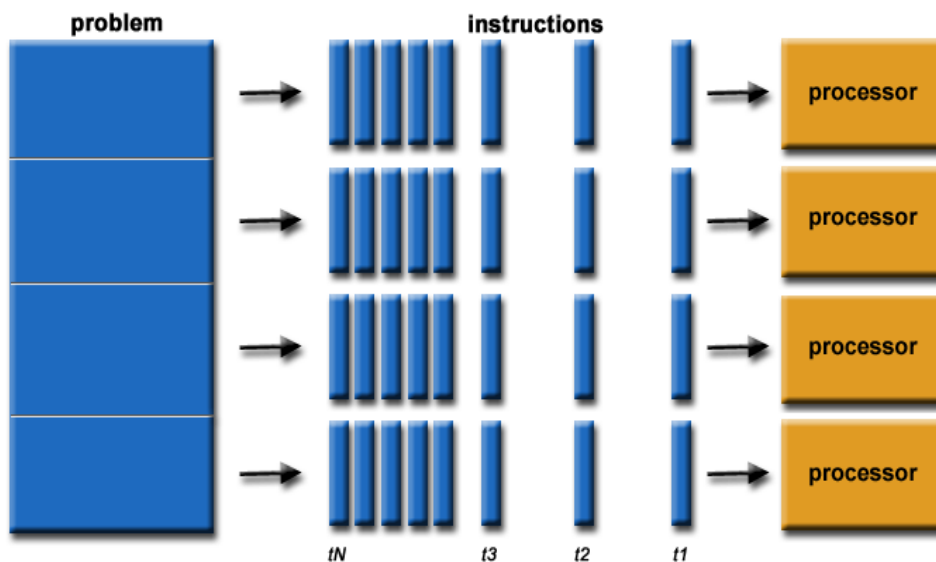
Παραδοσιακά, οι υπολογιστές βασίζονται τη λειτουργία τους σε ένα μοντέλο προγραμματισμού το οποίο είναι γνωστό ως *von Neuman* ή αλλιώς, ως ευρέως γνώριμο σειριακό, σύμφωνα με το οποίο κάθε εντολή εκτελείται σε σειρά, διαδοχικά, η μία μετά την άλλη όπως ορίζεται από τον προγραμματισμό. Η κεντρική μονάδα επεξεργασίας (*CPU*) που διαθέτει ο υπολογιστής επικοινωνεί με το τμήμα της μνήμης εκτελώντας εντολές τη μία μετά την άλλη διαδοχικά, με τη σειρά που εμφανίζονται στο πρόγραμμα που εκτελείται (εικόνα 1). Για την εκτέλεση της επόμενης εντολής απαραίτητη προϋπόθεση αποτελεί η ολοκλήρωση της προηγούμενης έως ότου επιλυθεί το πρόβλημα που έχει ανατεθεί.



Εικόνα 1 - Σειριακή επεξεργασία δεδομένων

Εξέλιξη της σειριακής επεξεργασίας αποτελεί η παράλληλη επεξεργασία δεδομένων που πρωταρχικό στόχο έχει την αύξηση της ταχύτητας του συστήματος. Στη βασική της μορφή προϋποθέτει την ταυτόχρονη χρήση πολλαπλών υπολογιστικών πόρων (επιπλέον του ενός επεξεργαστή ή ενός επεξεργαστή με πολλούς πυρήνες) για την επίλυση ενός προβλήματος. Το

πρόβλημα που καλείται να αντιμετωπίσει ο υπολογιστής χωρίζεται σε τμήματα για να μπορούν να επιλυθούν ταυτόχρονα. Στη συνέχεια, κάθε ένα τμήμα του προβλήματος αναλύεται περαιτέρω σε μία σειρά από οδηγίες. Οι εντολές κάθε τμήματος του προβλήματος μπορούν να εκτελεστούν ταυτόχρονα σε διαφορετικές CPUs (εικόνα 2). Η παράλληλη επεξεργασία δεδομένων απαιτεί στενή συνεργασία μεταξύ των επεξεργαστών για την επίλυση ενός κοινού προβλήματος, στόχο της οποίας, όπως προαναφέραμε, αποτελεί η επιτάχυνση της εργασίας και κατ' επέκταση η καλύτερη απόδοση. Για την επίτευξη του στόχου θα πρέπει οι επεξεργαστές, που συνεργάζονται από κοινού για την επίλυση του προβλήματος, να βρίσκονται πολύ κοντά ο ένας στον άλλο (αναφέρονται και ως *σφιχτά συνδεδεμένοι / tightly coupled*), συχνά και πάνω στην ίδια μητρική κάρτα ή ακόμα και στο ίδιο chip, αν πρόκειται για πολυπύρηνους επεξεργαστές. Όσο πιο κοντά βρίσκονται οι επεξεργαστές μεταξύ τους τόσο ελαχιστοποιείται ο χρόνος επικοινωνίας ανάμεσά τους, επιτρέποντας στο δίκτυο διασύνδεσης να έχει τυπικά πολύ μεγάλη ταχύτητα.

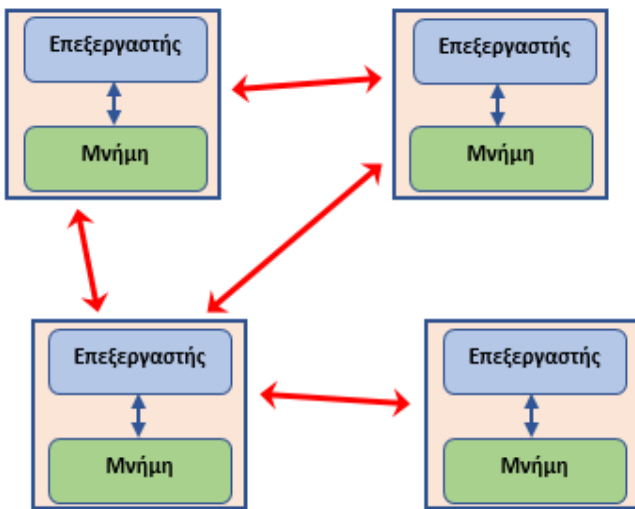


Εικόνα 2 - Παράλληλη επεξεργασία δεδομένων

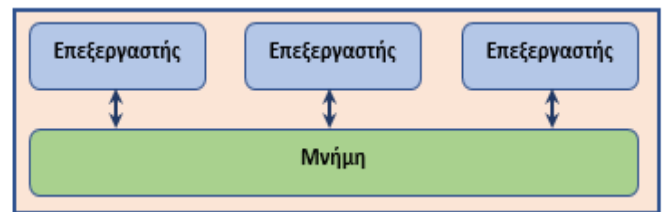
## 2.2 Κατανεμημένη επεξεργασία δεδομένων

Ένα υποσύνολο της παράλληλης επεξεργασίας δεδομένων αποτελεί η κατανεμημένη επεξεργασία στην οποία όλοι οι επεξεργαστές έχουν τις δικές τους μνήμες, διαφορετικούς όμως χώρους διευθύνσεων (είναι δηλαδή ανεξάρτητοι υπολογιστές συνδεδεμένοι σε ένα δίκτυο), και διαχειρίζονται από κοινού ένα σύνολο δεδομένων (εικόνες 3 και 4). Η κατανεμημένη επεξεργασία έχει ομοιότητες σε αρκετά σημεία με την παράλληλη, σε βαθμό τέτοιο που οι δύο όροι να δημιουργούν σύγχυση ως προς τις διαφορές τους. Ιδιαίτερα με την ανάπτυξη γρήγορων δικτύων επικοινωνίας τα όρια μεταξύ κατανεμημένης και παράλληλης επεξεργασίας έχουν γίνει δυσδιάκριτα. Χαρακτηριστική είναι η περίπτωση των συστοιχιών (*clusters*) όπου γεωγραφικά απομακρυσμένοι συνεργαζόμενοι υπολογιστές μπορούν να είναι στενά συζευγμένοι ανταλλάσσοντας δεδομένα, όπως σε ένα παράλληλο υπολογιστικό σύστημα. Όπως συμβαίνει και

στην παράλληλη επεξεργασία πολλοί υπολογιστές επικοινωνούν μεταξύ τους και ανταλλάσσουν πληροφορίες, με σκοπό τη λύση ενός προβλήματος. Σε αντίθεση, όμως, με την παράλληλη επεξεργασία, οι αποστάσεις μεταξύ των υπολογιστών είναι συνήθως ιδιαίτερα μεγάλες, με τη σύνδεσή τους να πραγματοποιείται με τυποποιημένα εμπορικά δίκτυα (π.χ. γραμμές VDSL, fiber) ή και ασύρματα (μέσω των τηλεπικοινωνιών). Οι υπολογιστές αυτοί είναι σε μεγάλο βαθμό ανεξάρτητοι και αναφέρονται συνήθως ως *χαλαρά συζευγμένοι (loosely coupled)*. Η ανάγκη για μεταξύ τους επικοινωνία είναι λιγότερο συχνή σε σχέση με τους υπολογιστές ενός παράλληλου συστήματος διαχείρισης δεδομένων, ενώ και οι διεργασίες διεκπεραιώνονται από προγράμματα που αναπτύχθηκαν ανεξάρτητα. Ακόμα και ένας παράλληλος υπολογιστής ή ένας πολυπύρηνος επεξεργαστής αποτελούν από μόνα τους ένα κατανεμημένο σύστημα διαχείρισης δεδομένων, το οποίο βρίσκεται σε μία κεντρική μονάδα και αποτελείται από πολλούς κόμβους (επεξεργαστές και μνήμες) οι οποίοι επικοινωνούν μέσω ενός γρήγορου δικτύου. Αν και η ταχύτητα αποτελεί το βασικό κριτήριο αξιολόγησης ενός παράλληλου συστήματος, στην κατανεμημένη διαχείριση δεδομένων υπάρχουν και άλλα λοιπά κριτήρια που καθορίζουν την αποτελεσματικότητα και αποδοτικότητα του συστήματος τα οποία θα παρουσιαστούν στη συνέχεια.



Εικόνα 4 - Λειτουργία κατανεμημένου συστήματος



Εικόνα 3 - Λειτουργία παράλληλου συστήματος

## 2.3 Παράλληλο και κατανεμημένο σύστημα διαχείρισης δεδομένων

Αν θα μπορούσαμε να ορίσουμε εν συντομία τι είναι ένα παράλληλο και κατανεμημένο σύστημα διαχείρισης δεδομένων θα λέγαμε ότι είναι ένα σύνολο το οποίο αποτελείται από ανεξάρτητες αλληλοσυνδεδεμένες οντότητες (*servers, clusters, workstations*, κ.λπ.), όπου η επικοινωνία τους πραγματοποιείται μέσω ενός δικτύου δεδομένων μεγάλης ταχύτητας, σε δίκτυα τοπικά ή ευρείας περιοχής (*WANs, LANs*), και συνεργάζονται από κοινού με σκοπό την επίλυση ενός προβλήματος, το οποίο θα ήταν αν όχι αδύνατον, τουλάχιστον εξαιρετικά δύσκολο να επιλυθεί μεμονωμένα. Ο παραπάνω ορισμός, θα πρέπει να επισημάνουμε ότι, αναφέρεται τόσο στο υλικό (*hardware*) όσο

και στο λογισμικό (*software*) που χρησιμοποιείται. Τα κατανεμημένα συστήματα διαχείρισης δεδομένων προσφέρουν ισχυρά κίνητρα χρήσης έναντι των κεντρικών συστημάτων. Ενδεικτικά αναφέρουμε τα σημαντικότερα:

1. **Εγγενείς κατανεμημένοι υπολογισμοί:** Σε πολλές εφαρμογές, όπως είναι η επίτευξη συναίνεσης μεταξύ τμημάτων ενός συνόλου που βρίσκονται γεωγραφικά σε μεγάλη απόσταση (π.χ. η μεταφορά χρημάτων στον τραπεζικό τομέα), οι υπολογισμοί πραγματοποιούνται εγγενώς.
2. **Κοινή χρήση πόρων:** Πόροι όπως πλήρη σύνολα δεδομένων, περιφερειακά, δεδομένα (αρχεία/μεταβλητές), βιβλιοθήκες δεν είναι δυνατόν να αναπαραχθούν πλήρως σε όλους τους ιστότοπους επειδή συχνά δεν είναι αποδοτικό αλλά ούτε και πρακτικό. Επιπλέον, δεν μπορούν να τοποθετηθούν σε μία μόνο ιστοσελίδα καθώς η ενέργεια αυτή θα δημιουργήσει το ανεπιθύμητο φαινόμενο *bottleneck* (δηλαδή ένας πόρος να μην μπορεί να αποδώσει τα μέγιστα γιατί περιορίζεται από την αργή επίδοση ενός άλλου). Συνεπώς, τέτοιοι πόροι τυπικά κατανέμονται σε όλο το σύστημα. Για παράδειγμα, κατανεμημένες βάσεις δεδομένων τύπου *DB2* διανέμουν τα δεδομένα τους σε διακομιστές, αντί να τα τοποθετήσουν σε μερικές ιστοσελίδες κερδίζοντας αξιοπιστία και γρήγορη πρόσβαση.
3. **Πρόσβαση σε γεωγραφικά απομακρυσμένα δεδομένα και πόρους:** Σε πολλές περιπτώσεις τα δεδομένα δεν μπορούν να αναπαραχθούν σε κάθε ιστότοπο που συμμετέχει στην κατανεμημένη εκτέλεση, είτε λόγω μεγάλου όγκου, είτε λόγω των ευαίσθητων στοιχείων που περιλαμβάνουν. Όπως, για παράδειγμα, τα δεδομένα μισθοδοσίας μίας πολυεθνικής εταιρείας ή οργανισμού τα οποία είναι και μεγάλα σε όγκο αλλά και ευαίσθητα. Συνεπώς, αποθηκεύονται σε έναν κεντρικό διακομιστή στον οποίο μπορεί να υποβληθεί ερώτημα από οποιοδήποτε τμήμα ή γραφείο. Επιπλέον ώθηση και πρόσθετη βαρύτητα έχει δοθεί στη διαχείριση των γεωγραφικά απομακρυσμένων πόρων και δεδομένων, λόγω της προόδου που έχει επιτευχθεί την τελευταία εικοσαετία στο σχεδιασμό και την ανάπτυξη των ασύρματων κινητών συσκευών (κινητά τηλέφωνα, υπολογιστές ταμπλέτες, κ.α.), καθώς και λόγω της ευρείας αποδοχής και χρήσης τους.
4. **Αυξημένη αξιοπιστία:** Ένα κατανεμημένο σύστημα έχει έμφυτη τη δυνατότητα να παρέχει αυξημένη αξιοπιστία λόγω της δυνατότητας του να αντιγράφει τα διαθέσιμα δεδομένα και τις εκτελέσιμες λειτουργίες. Επιπροσθέτως, αποτελεί πραγματικότητα το γεγονός ότι στα γεωγραφικά απομακρυσμένα κατανεμημένα συστήματα είναι δύσκολο να παρουσιαστεί ταυτόχρονη δυσλειτουργία υπό κανονικές συνθήκες. Ο όρος *αξιοπιστία* περιλαμβάνει τα ακόλουθα χαρακτηριστικά:
  - Διαθεσιμότητα: Τα δεδομένα θα πρέπει να είναι διαθέσιμα ανά πάσα στιγμή.
  - Ακεραιότητα: Η τιμή/κατάσταση των δεδομένων θα πρέπει να είναι σωστή και πλήρης κατά τη διάρκεια ταυτόχρονης πρόσβασης από πολλούς επεξεργαστές και σύμφωνη με τις οδηγίες που έχει ορίσει η εκάστοτε εφαρμογή.
  - Ανοχή σφαλμάτων: Η ικανότητα ανάκτησης των δεδομένων από αστοχίες του συστήματος.

5. **Αυξημένη σχέση απόδοσης/κόστους:** Με την κοινή χρήση πόρων και προσβασιμότητας σε γεωγραφικά απομακρυσμένα δεδομένα ο λόγος απόδοσης/κόστους αυξάνεται. Κάθε εργασία μπορεί να χωριστεί σε διάφορους υπολογιστές παρέχοντας υψηλότερες τιμές απόδοσης, χωρίς αντίστοιχη αύξηση του κόστους του συστήματος από τη χρήση των παράλληλων επεξεργαστών.

Επιπλέον, των παραπάνω κινήτρων, τα κατανεμημένα συστήματα προσφέρουν και τα ακόλουθα πλεονεκτήματα.

- **Επεκτασιμότητα (scalability).** Καθώς οι υπολογιστές συνήθως συνδέονται με ένα δίκτυο ευρείας περιοχής (WAN), η προσθήκη επιπλέον κόμβων δεν δημιουργεί άμεση συμφόρηση για το δίκτυο επικοινωνίας.
- **Προσαρμογή και σταδιακή επεκτασιμότητα.** Σε περιπτώσεις επέκτασης του συστήματος με επιπλέον κόμβους, οι υπολογιστές που φέρουν ετερογενείς επεξεργαστές μπορούν να προσαρμοστούν ευκολά στο σύστημα χωρίς να επηρεαστεί αρνητικά η απόδοση, εφόσον οι επεξεργαστές αυτοί εκτελούν τους ίδιους αλγορίθμους. Ομοίως, σε περιπτώσεις αναβάθμισης του συστήματος, υπάρχοντες επεξεργαστές παλαιότερης τεχνολογίας μπορούν εύκολα να αντικατασταθούν από άλλους σύγχρονους ισχυρότερους επεξεργαστές.

Παρά την πληθώρα των πλεονεκτημάτων και κινήτρων χρήσης των κατανεμημένων συστημάτων διαχείρισης δεδομένων, θα πρέπει να αναφερθούν και πιθανές δυσκολίες που πρέπει να προβλεφθούν και να αντιμετωπιστούν κατά τον σχεδιασμό και την υλοποίησή τους, οι οποίες είναι:

- **Αντιμετώπιση προβλημάτων.** Η διάγνωση προβλημάτων καθώς και η αντιμετώπισή τους είναι από τα σημαντικότερα μειονεκτήματα των κατανεμημένων συστημάτων διαχείρισης δεδομένων. Συχνότερα εξ αυτών αποτελούν η σύνδεση απομακρυσμένων κόμβων (*nodes*) και ο έλεγχος της ορθής επικοινωνίας τους, τα οποία θα πρέπει να αναλυθούν και να επιλυθούν.
- **Λογισμικό (software).** Λόγω της φύσης του λογισμικού, το οποίο είναι πιο πολύπλοκο, και περιλαμβάνει περισσότερα στοιχεία, έναντι ενός λογισμικού που λειτουργεί σε ένα κεντρικό σύστημα διαχείρισης δεδομένων, είναι μεγαλύτερη η πιθανότητα σφάλματος στη λειτουργία του.
- **Δίκτυο.** Είναι πιο ευάλωτο σε προβλήματα που προκύπτουν από το δίκτυο επικοινωνίας όπως υπερφόρτωση, απώλεια δεδομένων, δυσκολίες στη μετάδοση. Ως εκ τούτου, προβλήματα που αφορούν στην υποδομή του δικτύου αποτελούν και προβλήματα για το κατανεμημένο σύστημα διαχείρισης δεδομένων.
- **Ασφάλεια.** Η εύκολη πρόσβαση των χρηστών σε ένα κατανεμημένο υπολογιστικό σύστημα αυξάνει τον κίνδυνο ασφάλειας. Η κοινή χρήση των δεδομένων δημιουργεί πρόβλημα στην ασφάλεια τους.

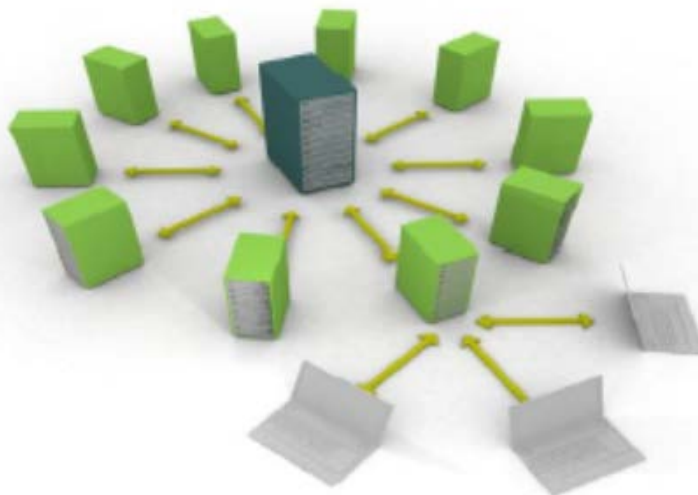
## 2.4 Εφαρμογές των καταναμημένων συστημάτων διαχείρισης δεδομένων

Σε δύο βασικές περιπτώσεις η χρήση καταναμημένων συστημάτων διαχείρισης δεδομένων θεωρείται επιβεβλημένη καθώς κεντροποιημένα συστήματα δεν μπορούν να λειτουργήσουν αποτελεσματικά. Η πρώτη αφορά στην περίπτωση που λόγω της φύσης της εφαρμογής που θα χρησιμοποιηθεί απαιτείται η χρήση ενός δικτύου επικοινωνίας για τη σύνδεση ενός συνόλου υπολογιστών, π.χ. όταν δεδομένα που έχουν παραχθεί σε μία φυσική τοποθεσία θα πρέπει να χρησιμοποιηθούν σε μια άλλη κοντινή ή απομακρυσμένη τοποθεσία. Η δεύτερη περίπτωση αφορά στο ενδεχόμενο που η χρήση πολλαπλών υπολογιστών είναι αρκετά πιο αποδοτική για πρακτικούς λόγους, σε σχέση με τη χρήση ενός υπολογιστή, π.χ. η διαχείριση μιας ομάδας υπολογιστών (*clusters*) χαμηλής απόδοσης (*low-end*) θα μπορούσε να έχει αρκετά λιγότερο κόστος σε σχέση με τη χρήση ενός υπολογιστή τελευταίας γενιάς (*high-end*) για να επιτευχθεί η επιθυμητή απόδοση.

Εφαρμογή των καταναμημένων υπολογιστικών συστημάτων μπορούμε να βρούμε σε πολλές σύγχρονες δραστηριότητες που σχετίζονται με υπολογιστικά και τηλεπικοινωνιακά συστήματα. Μερικά παραδείγματα που μπορούμε να αναφέρουμε είναι:

- **Στα δίκτυα τηλεπικοινωνιών:** Δίκτυα τηλεφωνίας και κινητής τηλεφωνίας, δίκτυα υπολογιστών όπως το internet, ασύρματα δίκτυα αισθητήρων, αλγόριθμους δρομολόγησης.
- **Σε εφαρμογές δικτύου:** Καταναμημένες βάσεις δεδομένων και καταναμημένα συστήματα βάσεων δεδομένων, στον παγκόσμιο ιστό (*World Wide Web*) και στα δίκτυα peer to peer (π.χ. Napster), στα online βιντεοπαιχνίδια όπου συμμετέχουν πολλοί παίκτες ταυτόχρονα, στα καταναμημένα συστήματα επεξεργασίας πληροφοριών, όπως είναι οι εταιρείες κρατήσεων αεροπορικών εισιτηρίων και τα τραπεζικά συστήματα.
- **Σε ελέγχους διαδικασιών σε πραγματικό χρόνο:** Βιομηχανικά συστήματα ελέγχου παραγωγής, σε συστήματα διαχείρισης εναέριας κυκλοφορίας αεροσκαφών, εταιρείες logistics και ηλεκτρονικού εμπορίου χρησιμοποιούν συστήματα παρακολούθησης σε πραγματικό χρόνο.
- **Για παράλληλους υπολογισμούς:** στην επιστημονική πληροφορική και στην καταναμημένη απόδοση σε γραφικά υπολογιστών. Η διάκριση που υπήρχε μεταξύ παράλληλων υπολογιστών και καταναμημένων συστημάτων δεν υφίσταται πλέον. Με την εξέλιξη των σύγχρονων λειτουργικών συστημάτων, των νέων επεξεργαστών και των υπηρεσιών νέφους (*cloud*), τα καταναμημένα υπολογιστικά συστήματα περιλαμβάνουν πλέον και την παράλληλη επεξεργασία.



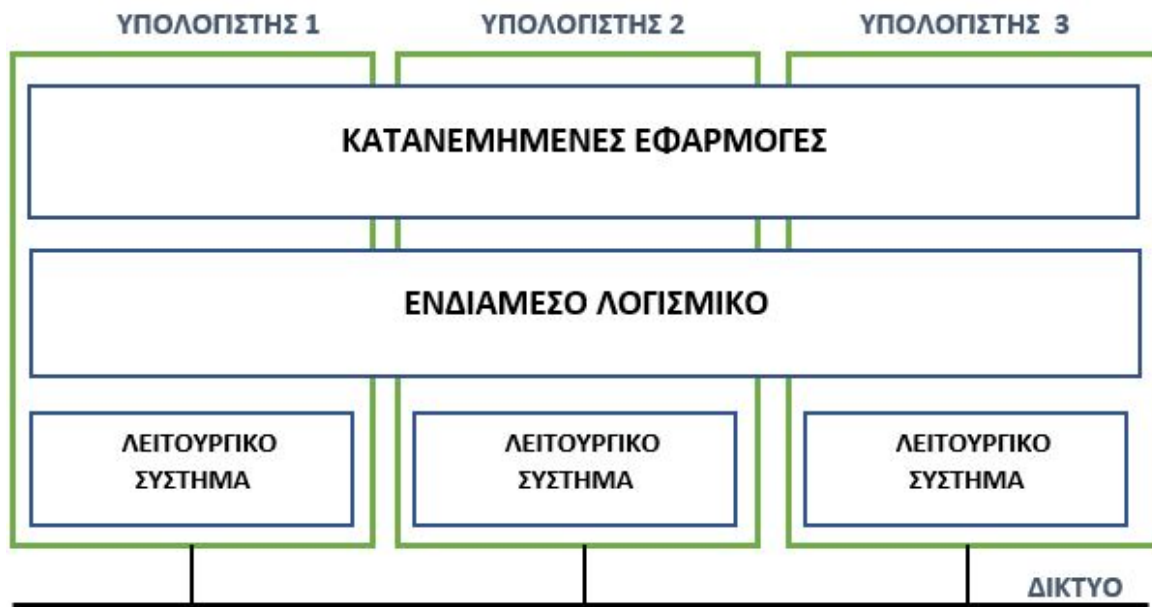


Εικόνα 5 – Αναπαράσταση τυπικού κατακεντρωμένου συστήματος (φυσικό επίπεδο)

## 2.5 Αρχιτεκτονική κατακεντρωμένων συστημάτων

Τα κατακεντρωμένα συστήματα δημιουργούνται πάνω σε υπάρχοντα δίκτυα επικοινωνίας και λειτουργικά συστήματα και περιλαμβάνουν μία ομάδα ανεξάρτητων υπολογιστών, οι οποίοι συνδέονται μέσω ενός δικτύου υπολογιστών με τη χρήση ενδιάμεσου λογισμικού (*middleware*). Το λογισμικό αυτό επιτρέπει στους υπολογιστές να μοιράζονται τους πόρους του συστήματος και να συντονίζουν τις δραστηριότητές τους, έτσι ώστε οι χρήστες να αντιλαμβάνονται το σύστημα ως ένα ολοκληρωμένο, ενιαίο σύστημα υπολογιστών (εικόνα 5). Αποτελεί ουσιαστικά τον ενδιάμεσο κρίκο που συνδέει κατακεντρωμένες εφαρμογές από διαφορετικές φυσικές τοποθεσίες με διαφορετικές πλατφόρμες υλικού, τεχνολογίες δικτύου επικοινωνίας, γλώσσες προγραμματισμού και λειτουργικά συστήματα και αναπτύσσεται βάσει συμφωνημένων προτύπων και πρωτοκόλλων (εικόνα 6). Παρέχει ουσιαστικές υπηρεσίες όπως ονομασία και έλεγχο ταυτότητας, διασφαλίζει ότι παράγονται ακριβή αποτελέσματα για ταυτόχρονες διαδικασίες, διανέμει τις εργασίες, παρέχει εξουσιοδότηση για τον καθορισμό δικαιωμάτων πρόσβασης σε πόρους, δεδομένα, ασφάλεια κ.λπ.

Ένα κατακεντρωμένο σύστημα ορίζεται σε φυσικό και λογικό επίπεδο από τα φυσικά και τα λογικά συστατικά του. Τα φυσικά συστατικά που αποτελούν το σύστημα περιλαμβάνουν ένα σύνολο υπολογιστών, που συνδέονται μεταξύ τους χωρίς να μοιράζονται τη μνήμη τους. Οι υπολογιστές είναι χαλαρά συνδεδεμένοι και επικοινωνούν χρησιμοποιώντας κατάλληλα πρωτόκολλα επικοινωνίας για τη μετάδοση μηνυμάτων μέσω του δικτύου επικοινωνίας. Το λογικά συστατικά του περιλαμβάνουν το λογισμικό από την πλευρά του χρήστη. Εμπεριέχουν το σύνολο των διαδικασιών καθώς και τα κανάλια επικοινωνίας μεταξύ των υπολογιστών.



Εικόνα 6 – Σχεδιασμός αρχιτεκτονικής καταναμημένου συστήματος (λογικό επίπεδο)

# 3

## Στόχοι της παράλληλης και κατανεμημένης διαχείρισης δεδομένων

Ο σχεδιασμός και η υλοποίηση ενός κατανεμημένου συστήματος διαχείρισης δεδομένων θα πρέπει να επιτύχει τέσσερις βασικούς στόχους για να εξασφαλιστεί αποτελεσματικότητα και αποδοτικότητα. Οι τέσσερις αυτοί στόχοι είναι: οι πόροι του συστήματος θα πρέπει να είναι εύκολα προσβάσιμοι, να αποκρύπτεται το γεγονός ότι οι πόροι είναι κατανεμημένοι σε ένα δίκτυο, να είναι ένα σύστημα “ανοικτό” και, τέλος, να είναι κλιμακούμενο(*scalable*). Ας δούμε πιο αναλυτικά τους τέσσερις αυτούς στόχους.

- 1) **Προσβασιμότητα πόρων:** Μια πολύ σημαντική επιδίωξη ενός κατανεμημένου συστήματος αποτελεί η εύκολη πρόσβαση και η κοινή χρήση όλων των πόρων του συστήματος. Ως πόρους εννοούμε οτιδήποτε τυπικά περιλαμβάνει το σύστημα, συσκευές αποθήκευσης, περιφερειακά, δεδομένα, αρχεία, υπηρεσίες, δίκτυα ,κ.λπ. Προφανή είναι τα οικονομικά κίνητρα επίτευξης ενός τέτοιου στόχου. Είναι οικονομικότερο π.χ. να υπάρχει μια αξιόπιστη *high end* εγκατάσταση αποθήκευσης των αρχείων, παρά ένας αποθηκευτικός χώρος για κάθε χρήστη ξεχωριστά. Παράλληλα, η σύνδεση χρηστών και πόρων διευκολύνει τη συνεργασία, καθώς και την ανταλλαγή πληροφοριών, όπως μπορούμε να διαπιστώσουμε άλλωστε και από τη μεγάλη επιτυχία του διαδικτύου όπου με τη χρήση απλών πρωτοκόλλων για την ανταλλαγή αρχείων, αλληλογραφίας, ήχου και βίντεο, καθώς και με τη χρήση ειδικού λογισμικού, έχει επιτραπεί σε γεωγραφικά απομακρυσμένες ομάδες ανθρώπων η δυνατότητα συνεργασίας.
- 2) **Διαφανής (*transparent*) διανομή πόρων και διαδικασιών:** Θα πρέπει το σύστημα να αποκρύπτει το γεγονός ότι οι πόροι καθώς και οι διαδικασίες του κατανέμονται φυσικά σε πολλούς υπολογιστές. Η διανομή των πόρων και των διαδικασιών θα πρέπει να είναι διαφανής, δηλαδή αόρατη, στις εφαρμογές και στους τελικούς χρήστες. Η έννοια της διαφάνειας μπορεί να εφαρμοστεί σε διάφορες πτυχές ενός κατανεμημένου συστήματος όπως παρουσιάζεται στον πίνακα παρακάτω.

Διαφάνεια	Περιγραφή
Πρόσβαση	Απόκρυψη τρόπου πρόσβασης σε ένα αντικείμενο
Τοποθεσία	Απόκρυψη τοποθεσίας ενός αντικειμένου
Μετατόπιση	Απόκρυψη στην περίπτωση που ένα αντικείμενο μετατοπιστεί κατά τη χρήση
Μετανάστευση	Απόκρυψη ενός αντικειμένου που θα μετακινηθεί σε άλλη τοποθεσία
Αναπαραγωγή	Απόκρυψη όταν ένα αντικείμενο αναπαράγεται
Συγχρονισμός	Απόκρυψη ότι ένα αντικείμενο μπορεί να κοινοποιηθεί από πολλούς ανεξάρτητους χρήστες
Αποτυχίας	Απόκρυψη της αστοχίας και επαναφοράς ενός αντικειμένου

- 3) **“Ανοικτό” σύστημα:** Ένα κατανεμημένο σύστημα θα πρέπει να είναι “ανοικτό”. Με τον όρο αυτό εννοούμε πως θα πρέπει το σύστημα να διασφαλίζει ότι κάθε υποσύστημα είναι συνεχώς ανοικτό σε αλληλεπίδραση με άλλα συστήματα. Βασικά χαρακτηριστικά του ανοικτού συστήματος είναι η διαλειτουργικότητα, η φορητότητα και η επεκτασιμότητα.
- Διαλειτουργικότητα: Χαρακτηρίζει το μέγεθος κατά το οποίο δύο υλοποιήσεις συστημάτων ή εξαρτημάτων από διαφορετικούς κατασκευαστές μπορούν να συνυπάρχουν και να συνεργαστούν βασιζόμενοι ο ένας στις υπηρεσίες του άλλου, όπως ορίζεται από ένα κοινό πρότυπο.
  - Φορητότητα: Καθορίζει σε πιο βαθμό αναπτύχθηκε μία εφαρμογή για ένα κατανεμημένο σύστημα  $A$  και πως μπορεί να εκτελεστεί χωρίς τροποποιήσεις σε ένα διαφορετικό, ετερογενές σύστημα  $B$  που εφαρμόζει τις ίδιες διεπαφές με το σύστημα  $A$ .
  - Επεκτασιμότητα: Προσδιορίζει την ευχέρεια με την οποία είναι δυνατόν να διαμορφωθεί το σύστημα από διαφορετικά στοιχεία. Επίσης, χαρακτηρίζει την αβίαστη προσθήκη νέων στοιχείων ή την αντικατάσταση υπαρχόντων χωρίς να επηρεάζουν αυτά που παραμένουν στη θέση τους. Για παράδειγμα, σε ένα επεκτάσιμο σύστημα θα πρέπει να δίνεται η δυνατότητα προσθήκης ή αντικατάστασης νέων στοιχείων, έστω και αν αυτά λειτουργούν σε διαφορετικό λειτουργικό σύστημα.
- 4) **Κλιμάκωση:** Η έννοια αυτή αναφέρεται στην ικανότητα ενός συστήματος να διατηρεί σταθερές κάποιες βασικές ιδιότητες και να λειτουργεί αποτελεσματικά, ακόμα και όταν πραγματοποιείται μία κλιμάκωση (πάντα προς τα πάνω) των πόρων και των χρηστών του συστήματος. Το ζήτημα της κλιμάκωσης αποτελεί μείζον θέμα στο σχεδιασμό και την ανάπτυξη κατανεμημένων συστημάτων. Στην ιδανική περίπτωση το σύστημα καθώς και το λογισμικό των εφαρμογών δε θα πρέπει να αλλάζει όταν αυξάνεται η κλίμακα του συστήματος, αν και αυτό είναι δύσκολο να επιτευχθεί. Ο βαθμός κλιμάκωσης ενός συστήματος μπορεί να εκτιμηθεί μέσα από τρεις διαστάσεις:

- Το σύστημα μπορεί να είναι κλιμακούμενο αναφορικά με το μέγεθος του, εννοώντας την ευκολία με την οποία μπορούμε να προσθέσουμε πόρους και χρήστες χωρίς να παρατηρηθεί σημαντική απώλεια της απόδοσης.
- Να είναι γεωγραφικά κλιμακούμενο, δηλαδή οι χρήστες και οι πόροι του συστήματος να βρίσκονται γεωγραφικά αρκετά απομακρυσμένοι μεταξύ τους. Εδώ θα πρέπει να επισημανθεί ότι δεν δίνεται ιδιαίτερη βαρύτητα στις καθυστερήσεις που οφείλονται στο δίκτυο επικοινωνίας.
- Να είναι διοικητικά κλιμακούμενο, δηλαδή να είναι εύκολη η διαχείριση, ακόμα και η διεύρυνση, του συστήματος που εκτείνεται σε περισσότερους από έναν ανεξάρτητους διοικητικούς οργανισμούς.

Ο σχεδιασμός κλιμακούμενων συστημάτων κατανεμημένης επεξεργασίας θα πρέπει να εστιάζεται στις ακόλουθες προκλήσεις:

- **Έλεγχος του κόστους των φυσικών πόρων:** Καθώς αυξάνεται η ζήτηση ενός πόρου, θα πρέπει με λογικό κόστος να είναι δυνατή η επέκταση του συστήματος για την κάλυψη των αναγκών. Σε γενικές γραμμές θα πρέπει να είναι επεκτάσιμο. Σε ένα σύστημα με αριθμό χρηστών  $n$ , η ποσότητα των πόρων που απαιτείται για την υποστήριξη τους να είναι  $O(n)$ , δηλαδή ανάλογο του  $n$ , π.χ. αν ένας διακομιστής αρχείων μπορεί να υποστηρίξει 30 χρήστες, τότε 2 διακομιστές θα πρέπει να υποστηρίξουν 60 χρήστες. Αν και φαίνεται ευκολά πραγματοποιηίσιμο, στην πράξη δεν είναι.
- **Έλεγχος της απώλειας απόδοσης:** Η αύξηση του μεγέθους ενός συστήματος έχει ως αποτέλεσμα τη μείωση της απόδοσης. Σε αλγορίθμους που χρησιμοποιούν ιεραρχικές δομές έχει παρατηρηθεί ότι κλιμακώνονται καλύτερα, σε σχέση με αυτούς που χρησιμοποιούν γραμμικές δομές. Εντούτοις, είναι αδύνατον η αύξηση του μεγέθους να μην επιφέρει κάποια απώλεια απόδοσης στο σύστημα.
- **Αποτροπή εξάντλησης των πόρων του λογισμικού:** Θα πρέπει να γίνεται μια λογική πρόβλεψη του μεγέθους και της ζήτησης που πρόκειται να λάβει μελλοντικά το σύστημα, αν και είναι δύσκολο να προβλεφθεί η ζήτηση που θα τεθεί στο σύστημα μερικά χρόνια αργότερα. Χρειάζεται, όμως, ιδιαίτερη προσοχή καθώς μία υπερβολική πρόβλεψη μελλοντικής χρήσης του συστήματος μπορεί να επιφέρει περισσότερα προβλήματα από μία επικείμενη προσαρμογή. Παράδειγμα έλλειψης επεκτασιμότητας αποτελεί και η δημιουργία των διαδικτυακών διευθύνσεων των υπολογιστών κατά τη δεκαετία του '70 όπου αποφασίστηκε να χρησιμοποιηθεί μέγεθος 32 bit ( $IPv4$ ). Ο αριθμός των παρεχόμενων διευθύνσεων άρχισε να εξαντλείται μερικές δεκαετίες αργότερα και χρειάστηκε να υιοθετηθεί νέο πρωτόκολλο μεγέθους 128 bit ( $IPv6$ ), απαιτώντας σημαντικές τροποποιήσεις σε πολλά στοιχεία λογισμικού. Φυσικά, ήταν αδύνατον να προβλεφθεί από τους σχεδιαστές η εκρηκτική αύξηση της χρήσης του διαδικτύου, δεν παύει όμως να αποτελεί παράδειγμα έλλειψης προνοητικότητας και επεκτασιμότητας.

- **Αποφυγή σημείων συμφόρησης:** Θα πρέπει ο σχεδιασμός των αλγορίθμων να είναι τέτοιος ώστε κοινόχρηστοι πόροι, που είναι πολύ συχνά προσβάσιμοι, να μην δημιουργούν συμφόρηση προκαλώντας μείωση της απόδοσης. Καλή τακτική αποτελεί η αποκέντρωση των αλγορίθμων, ενώ η αναπαραγωγή καθώς και η προσωρινή αποθήκευση μπορούν να χρησιμοποιηθούν για τη βελτίωση της απόδοσης των πόρων που χρησιμοποιούνται περισσότερο.

Ολοκληρώνοντας, θα πρέπει να τονιστεί ότι οι παραπάνω στόχοι οφείλουν να λειτουργούν συμπληρωματικά στην αδιαμφισβήτητη και θεμελιώδη επιδίωξη της παράλληλης και κατανεμημένης επεξεργασίας που αποτελεί η **ορθότητα** του αποτελέσματος ενός ερωτήματος, η οποία θα πρέπει να συντελείται βάσει αποδοτικών διεργασιών, που θα βασίζονται στην ελαχιστοποίηση της κατανάλωσης των πόρων του συστήματος και του χρόνου διεκπεραίωσης του.

# 4

## Διαμερισμός και τοποθέτηση των δεδομένων σε καταναμημένα συστήματα

Τα δεδομένα που καλείται να διαχειριστεί ένα καταναμημένο σύστημα διαχείρισης δεδομένων βρίσκονται σε βάσεις δεδομένων οι οποίες είναι αποθηκευμένες σε πόρους του συστήματος και πιθανότατα σε διαφορετικές τοποθεσίες. Οι βάσεις αυτές αναφέρονται ως καταναμημένες βάσεις δεδομένων (*distributed databases*) και ορίζονται ως ομάδες βάσεων δεδομένων οι οποίες είναι λογικά συνδεδεμένες και διάσπαρτες σε ένα δίκτυο υπολογιστών. Τη διαχείρισή τους αναλαμβάνει ένα λογισμικό που ορίζεται ως καταναμημένο σύστημα διαχείρισης δεδομένων (*Distributed Data Base System*), με τρόπο τέτοιο που κάνει τη κατανομή της διαφανή (αόρατη) στους χρήστες, δίνοντάς τους την εντύπωση πως διαχειρίζονται μία ενιαία βάση. Μία καταναμημένη βάση δεδομένων δεν είναι απλά μία ομάδα τοπικών βάσεων που επικοινωνούν μεταξύ τους και είναι τυχαία διάσπαρτες. Υπάρχει μεταξύ τους μία λογική συνοχή, χωρίς ωστόσο η εσωτερική τους δομή, όπως αναφέρθηκε, να είναι εμφανής στους χρήστες. Βασικά γνωρίσματά της αποτελούν τα παρακάτω:

- Ανεξαρτησία από τα δεδομένα: Η φυσική οργάνωση των δεδομένων είναι διαφανής στον προγραμματιστή.
- Έλεγχος του συστήματος: Πραγματοποιείται τοπικά από τοπικούς διαχειριστές (*administrators*) και καθολικά από κάποιον επιβλέποντα διαχειριστή.
- Ο έλεγχος συντονισμού: Η επικοινωνία των συστημάτων, η αξιοπιστία καθώς και ο ορισμός δικαιωμάτων στους χρήστες αποτελούν βασικά χαρακτηριστικά και των DDBS.
- Ύπαρξη αντιγράφων: Σε ένα DDBS η ύπαρξη πλεονασμού των δεδομένων (σε αντίθεση με τις μη-καταναμημένες βάσεις), όχι μόνο δεν θεωρείται πρόβλημα, αντιθέτως, η ύπαρξη τοπικών αντιγράφων των πληροφοριών είναι έως ένα βαθμό και επιδιωκόμενη, καθώς προσφέρει αύξηση της απόδοσης του συστήματος, ευνοεί τη συντήρηση του συστήματος, ελαχιστοποιεί τον κίνδυνο απώλειας των δεδομένων σε περιπτώσεις αστοχίας και μειώνει το φόρτο μεταφοράς δεδομένων του δικτύου επικοινωνιών.

Σύμφωνα με τα παραπάνω γνωρίσματα οι βασικότερες προκλήσεις που θα πρέπει να αντιμετωπιστούν σε περιβάλλοντα καταναμημένων βάσεων δεδομένων σχετίζονται με το πρόβλημα της σχεδίασης, του ελέγχου συντονισμού και της βελτιστοποίησης των ερωτήσεων αλγορίθμων. Σε ένα καταναμημένο σύστημα οι σχέσεις αποθηκεύονται σε διάφορες τοποθεσίες. Αν οι σχέσεις αυτές αποθηκεύονταν σε μία μόνο τοποθεσία, αυτό θα συνεπάγονταν με αυξημένο κόστος μετάδοσης των μηνυμάτων. Για τον περιορισμό του προβλήματος αυτού μπορεί μία μεμονωμένη σχέση να κατακερματιστεί (*fragmentation*) και τα επιμέρους τμήματα να αποθηκευτούν σε

τοποθεσίες οι οποίες είναι πιο συχνά προσπελάσιμες ή να διαμοιραστεί σε αντίγραφα (replication), σε θέσεις όπου η σχέση έχει μεγαλύτερη ζήτηση. Στις περισσότερες των περιπτώσεων ένας συνδυασμός και των δύο τεχνικών χρησιμοποιείται ως πιο αποτελεσματική λύση.

## 4.1 Κατακερματισμός των δεδομένων (fragmentation)

Όπως αναφέρθηκε και παραπάνω, η κατανεμημένη βάση δεδομένων είναι μία λογικά αλληλένδετη συλλογή δεδομένων που διανέμεται φυσικά σε ένα δίκτυο υπολογιστών. Αυτό σημαίνει ότι τα δεδομένα είναι μοιρασμένα μέσα στο δίκτυο και μέσα στα DDBS. Για να επιτευχθεί όμως ο στόχος των κατανεμημένων συστημάτων που είναι η διαφάνεια διανομής των δεδομένων, αλλά και η διάθεση των απαιτούμενων στοιχείων στους χρήστες από την πλησιέστερη τοποθεσία όσο το δυνατόν ταχύτερα, θα πρέπει τα δεδομένα αυτά να κατακερματιστούν περαιτέρω. Ο επιπρόσθετος αυτός κατακερματισμός παρέχει τη δυνατότητα μη αποθήκευσης των δεδομένων, τα οποία δεν είναι απαραίτητα για τη λειτουργία τοπικών εφαρμογών, μειώνοντας έτσι τον όγκο των αποθηκευμένων στοιχείων. Ως εκ τούτου, όλα τα δεδομένα ενός πίνακα χωρίζονται και αποθηκεύονται στα DDBS σύμφωνα με τις απαιτήσεις του χρήστη ή ανάλογα με την τοποθεσία τους. Πιο αναλυτικά, θα μπορούσαμε να αναφέρουμε τους παρακάτω λόγους για τους οποίους ο κατακερματισμός των δεδομένων θεωρείται εξαιρετικά σημαντικός για τα κατανεμημένα συστήματα διαχείρισης δεδομένων:

- Αυξάνει την αποτελεσματικότητα των ερωτημάτων μειώνοντας το μέγεθος μιας σχέσης σε υποσύνολα. Συνεπώς, η πρόσβαση στα δεδομένα της απαιτεί λιγότερο χρόνο στο δίκτυο.
- Παρέχει εύκολη πρόσβαση στα δεδομένα, καθώς σε αυτά που απαιτείται πιο συχνή πρόσβαση τοποθετούνται πλησιέστερα στο χρήστη.
- Παρέχει ασφάλεια στα δεδομένα καθώς μόνο χρήσιμες και έγκυρες εγγραφές είναι διαθέσιμες στο χρήστη. Το τμήμα του DDBS που βρίσκεται κοντά στο χρήστη δεν περιέχει ανεπιθύμητα δεδομένα αλλά μόνο αυτά που πραγματικά χρειάζεται.
- Αυξάνει τον παραλληλισμό. Επιτρέπει στους χρήστες να έχουν πρόσβαση στην ίδια σχέση από διαφορετικές τοποθεσίες. Οι χρήστες από διαφορετικές τοποθεσίες μπορούν να έχουν πρόσβαση στον ίδιο πίνακα από τη δικιά τους τοποθεσία βλέποντας τα δεδομένα που προορίζονται γι' αυτούς. Διαφορετικά, αν όλοι οι χρήστες έχουν πρόσβαση στο πίνακα από μια τοποθεσία θα πρέπει να περιμένουν στη σειρά για να εκτελεστεί το ερώτημά τους.
- Ισορροπημένη αποθήκευση των δεδομένων. Αυτά κατανέμονται ομοιόμορφα μεταξύ των βάσεων δεδομένων της κατανεμημένης βάσης.
- Αυξάνει την αξιοπιστία ανάκτησης των δεδομένων. Αν όλοι οι χρήστες από διαφορετικές τοποθεσίες έχουν πρόσβαση σε ενιαία βάση δεδομένων, τότε υπάρχει τεράστια αύξηση φόρτου του δικτύου. Αυτή η εκτίναξη μεταφοράς δεδομένων δεν εγγυάται την ανάκτηση και την επιστροφή σωστών εγγραφών στον χρήστη. Η πρόσβαση σε ένα μόνο κομμάτι της βάσης, αυτής που βρίσκεται πλησιέστερα του, μειώνει τον κίνδυνο απώλειας των δεδομένων μεγιστοποιώντας στο έπακρο το ποσοστό λήψης ορθών εγγραφών.



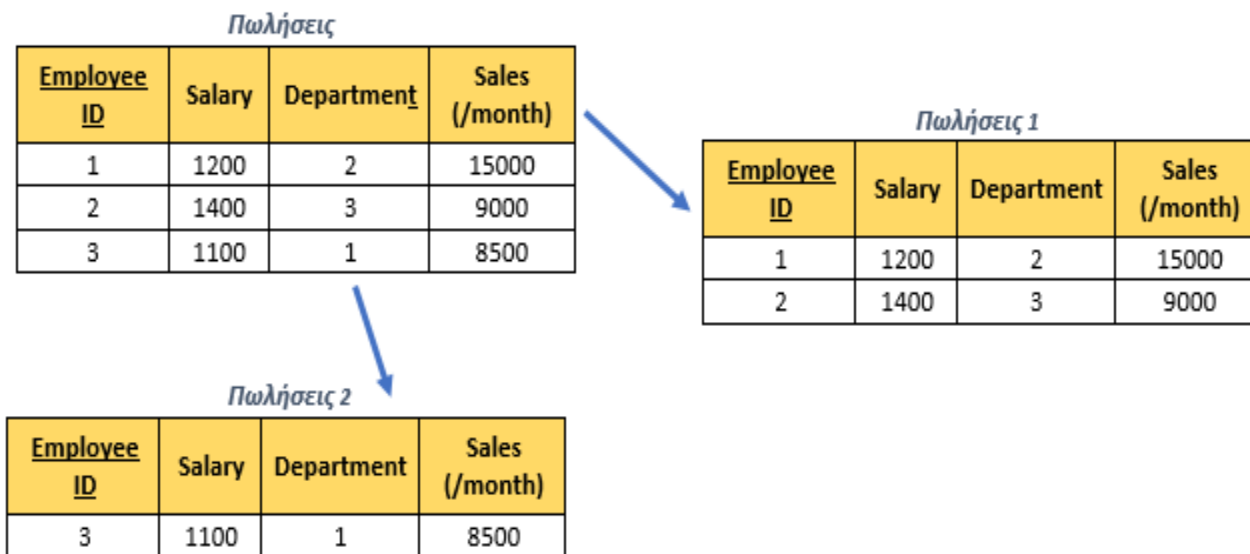
Ωστόσο, ο κατακερματισμός των δεδομένων δεν μπορεί να πραγματοποιηθεί τυχαία. Θα πρέπει να πληρούνται οι ακόλουθες ιδιότητες οι οποίες πιστοποιούν ότι η κατακερμασμένη βάση δεδομένων δεν υφίσταται σημασιολογική αλλαγή κατά τον κερματισμό. Δηλαδή, θα πρέπει σε περίπτωση ενοποίησης των τμημάτων μίας σχέσης να μας δίνουν το ίδιο αποτέλεσμα με την αρχική σχέση πριν τον κατακερματισμό.

- ✓ **Πληρότητα (completeness):** Κατά τη διάρκεια τεμαχισμού ενός τμήματος του DDBS δε θα πρέπει να λαμβάνονται υπόψη μεμονωμένες εγγραφές του πίνακα. Ο κατακερματισμός θα πρέπει να πραγματοποιείται σε δεδομένα ολόκληρου του πίνακα για να έχει το σωστό αποτέλεσμα. Έτσι, διασφαλίζεται ότι όλα τα δεδομένα σε μία σχέση κατανέμονται σε κομμάτια χωρίς απώλειες, π.χ. εάν θέλουμε να κατακερματίσουμε έναν πίνακα με όνομα ΠΩΛΗΣΕΙΣ, τότε θα πρέπει να εξετάσουμε ολόκληρο τον πίνακα ΠΩΛΗΣΕΙΣ για να φτιάξουμε τμήματα του. Δε θα πρέπει να δημιουργηθεί υποσύνολο εγγραφών του πίνακα ΠΩΛΗΣΕΙΣ.
- ✓ **Συνθήκη ανασύνθεσης (reconstruction):** Όταν ανασυνδεθούν όλα τα τμήματα, αυτά θα πρέπει να παρέχουν τα δεδομένα και τις σχέσεις ολόκληρου του πίνακα. Δηλαδή, ολόκληρος ο πίνακας θα πρέπει να είναι σε θέση να ανακατασκευαστεί χρησιμοποιώντας όλα τα τμήματά του, π.χ. όταν όλα τα τμήματα του πίνακα ΠΩΛΗΣΕΙΣ της βάσης δεδομένων συνδυαστούν θα πρέπει να παρέχουν πλήρεις εγγραφές του πίνακα ΠΩΛΗΣΕΙΣ.
- ✓ **Συνθήκη διαχωρισμού (disjointness):** Δε θα πρέπει να υπάρχουν επικαλυπτόμενα δεδομένα στα τμήματα γιατί θα είναι δύσκολο να διατηρηθεί η συνέπεια των δεδομένων. Θα πρέπει να καταβληθεί προσπάθεια για να δημιουργηθεί η ίδια αναπαραγωγή σε όλα τα αντίγραφα, π.χ. εάν έχουμε τμήματα στον πίνακα ΠΩΛΗΣΕΙΣ, με βάση την τοποθεσία, τότε δε θα πρέπει να υπάρχουν δύο τμήματα με τα στοιχεία του ίδιου υπαλλήλου.

## 4.2 Αρχιτεκτονικές κατακερματισμού των δεδομένων

Δύο είναι οι γενικές αρχιτεκτονικές κατακερματισμού των δεδομένων που μπορούν να εφαρμοστούν, ο *οριζόντιος* και ο *κάθετος* διαμερισμός. Επιπλέον, υπάρχει και η δυνατότητα ενός συνδυασμένου κατακερματισμού των δεδομένων χρησιμοποιώντας τους δύο γενικούς τύπους, ο *υβριδικός* κατακερματισμός.

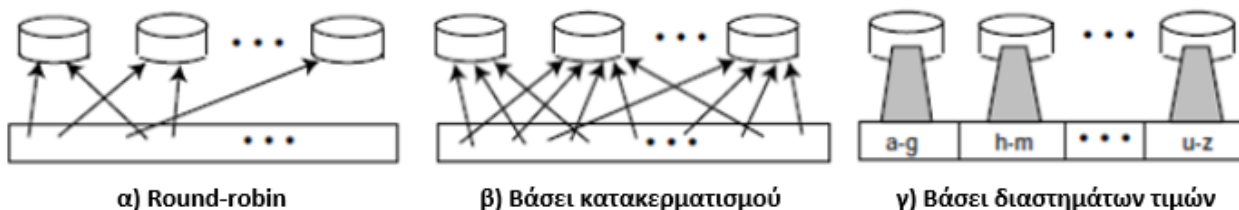
**Οριζόντιος κατακερματισμός:** Ο οριζόντιος τεμαχισμός διαχωρίζει μία σχέση κατά μήκος των πλειάδων της. Συνεπώς, κάθε τμήμα περιλαμβάνει ένα υποσύνολο των πλειάδων της σχέσης (εικόνα 7). Για τον οριζόντιο κατακερματισμό υπάρχουν δύο διαφορετικές εκδοχές, ο *πρωταρχικός* οριζόντιος κατακερματισμός όπου ο τεμαχισμός μιας σχέσης πραγματοποιείται χρησιμοποιώντας τις κατηγορίες που καθορίζονται σε αυτή τη σχέση και ο *συμπληρωματικός* οριζόντιος κατακερματισμός, όπου ο διαχωρισμός μιας σχέσης απορρέει από τον καθορισμό κατηγοριών σε άλλη σχέση. Η επαλήθευση του σωστού κατακερματισμού επιβεβαιώνεται με τη συνένωση των οριζόντιων τμημάτων που θα πρέπει να είναι ίση με την αρχική σχέση.



Εικόνα 7 - Παράδειγμα αρχιτεκτονικής οριζοντίου κατακερματισμού

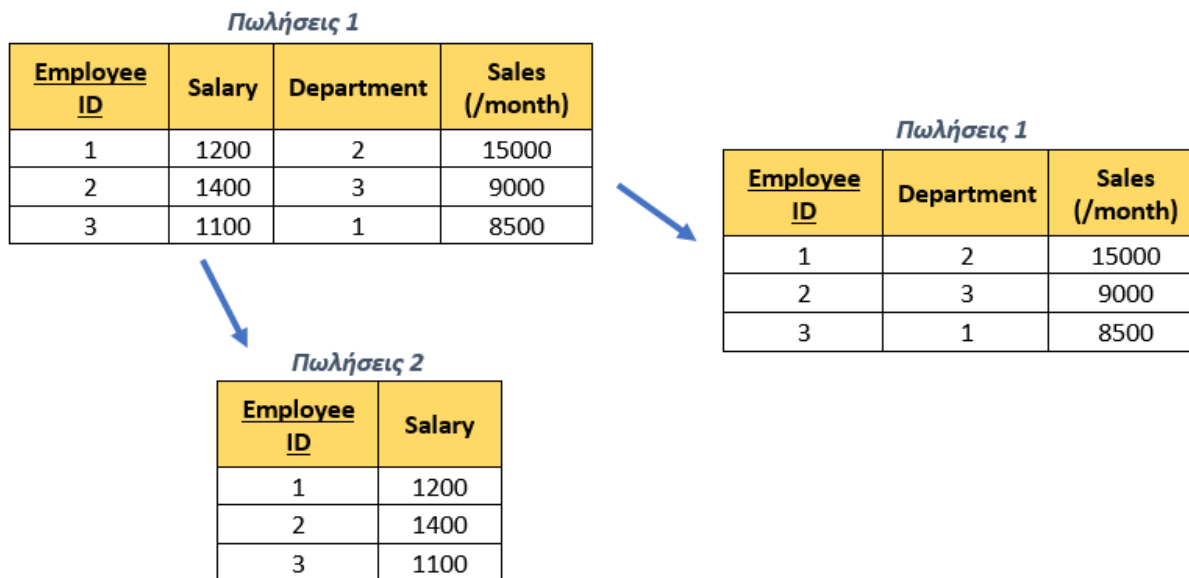
Κατακερματίζοντας ένα μεγάλο σύνολο δεδομένων οριζόντια σε πολλούς δίσκους παρέχεται η δυνατότητα στο κατακερματισμένο σύστημα να εκμεταλλευτεί τη μέγιστη δυνατή ταχύτητα προσπέλασης (*I/O bandwidth*) από και προς τους δίσκους της βάσης δεδομένων διαβάζοντας και εγγράφοντας παράλληλα. Οι πιο κοινές τεχνικές οριζοντίου διαμερισμού είναι:

- Round-robin: Είναι ο απλούστερος τρόπος και εξασφαλίζει την ομοιόμορφη κατανομή δεδομένων (εικόνα 8α). Έχοντας  $n$  κατατιμήσεις η πλειάδα  $i$  αντιστοιχείται με κόμβο, τον κόμβο  $i \bmod n$ . Αυτή η στρατηγική επιτρέπει σε μία διαδοχική σχέση να γίνει παράλληλα. Ωστόσο, η άμεση πρόσβαση σε μεμονωμένες πλειάδες απαιτεί πρόσβαση σε ολόκληρη τη σχέση.
- Βάσει κατακερματισμού (hash): Εφαρμόζεται σε επιλεγμένα πεδία μίας πλειάδας σύμφωνα με την τιμή κάποιου πεδίου (εικόνα 8β). Εδώ επιτρέπεται η ακριβής αντιστοίχιση των ερωτημάτων με την επιλογή /χαρακτηριστικό που θέλουμε να επεξεργαστούμε.
- Βάσει διαστημάτων τιμών (range): Ο διαμερισμός των πλειάδων γίνεται βάσει του εύρους τιμών κάποιου χαρακτηριστικού (εικόνα 8γ). Εκτός από την υποστήριξη ερωτημάτων εύρους τιμών είναι κατάλληλο και για ερωτήματα ακριβής αντιστοίχισης (όπως και στο hashing).



Εικόνα 8 - Τεχνικές οριζοντίου κατακερματισμού

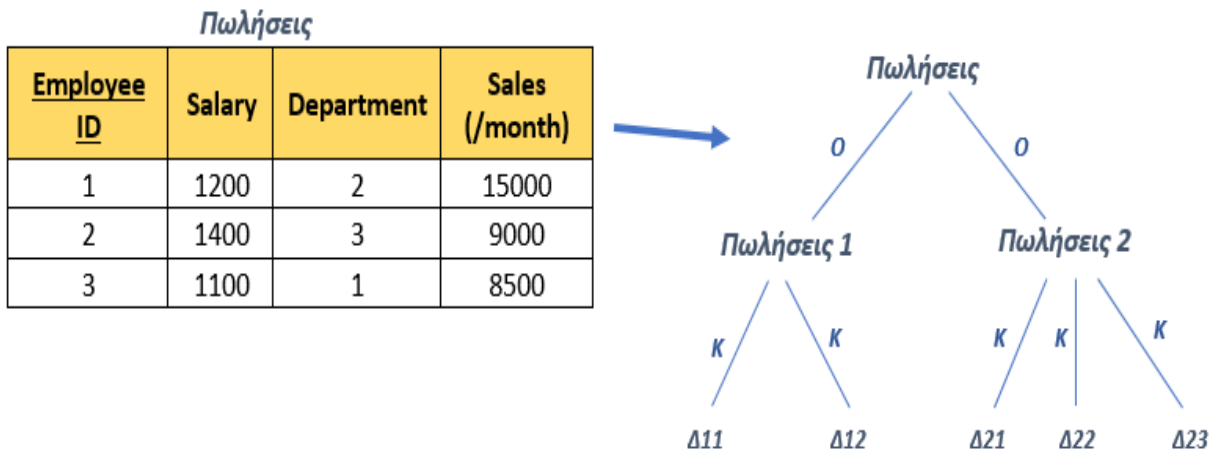
**Κάθετος κατακερματισμός:** Εδώ το κάθε τμήμα αποτελείται από ένα υποσύνολο στηλών της αρχικής σχέσης (εικόνα 9). Επιδίωξη του κάθετου κατακερματισμού αποτελεί ο τεμαχισμός μίας σχέσης σε ένα υποσύνολο από μικρότερες σχέσεις, έτσι ώστε πολλές εφαρμογές του χρήστη να εκτελούνται μόνο από ένα τμήμα. Βέλτιστος κάθετος κατακερματισμός θεωρείται αυτός που καταφέρνει να ελαχιστοποιήσει το χρόνο εκτέλεσης των εφαρμογών που εκτελούνται σε αυτά τα τμήματα. Η συνένωση των κατακόρυφων τμημάτων θα πρέπει να συγκροτεί μια σύνθεση που δε θα έχει απώλεια πληροφορίας της αρχικής σχέσης.



Εικόνα 9 - Παράδειγμα αρχιτεκτονικής κάθετου κατακερματισμού

**Υβριδικός ή μεικτός κατακερματισμός:** Σε αρκετές περιπτώσεις ένας οριζόντιος ή κάθετος κατακερματισμός δεν είναι αρκετός για να καλύψει τις απαιτήσεις των εφαρμογών του χρήστη. Σε τέτοια ενδεχόμενα μπορεί ένας κάθετος κατακερματισμός να ακολουθείται από έναν οριζόντιο ή και το αντίστροφο, δημιουργώντας έτσι ένα δέντρο κατακερματισμού στο οποίο η ρίζα αντιστοιχεί στην καθολική σχέση, τα φύλλα στα καταληκτικά τμήματα και οι μεσαίοι κόμβοι στα ενδιάμεσα στάδια κατακερματισμού (εικόνα 10). Δεδομένου ότι και οι δύο γενικοί τύποι εφαρμόζονται συνδυαστικά για την κατάτμηση, ο ένας μετά τον άλλον, ο εναλλακτικός αυτός τρόπος ονομάζεται υβριδικός.

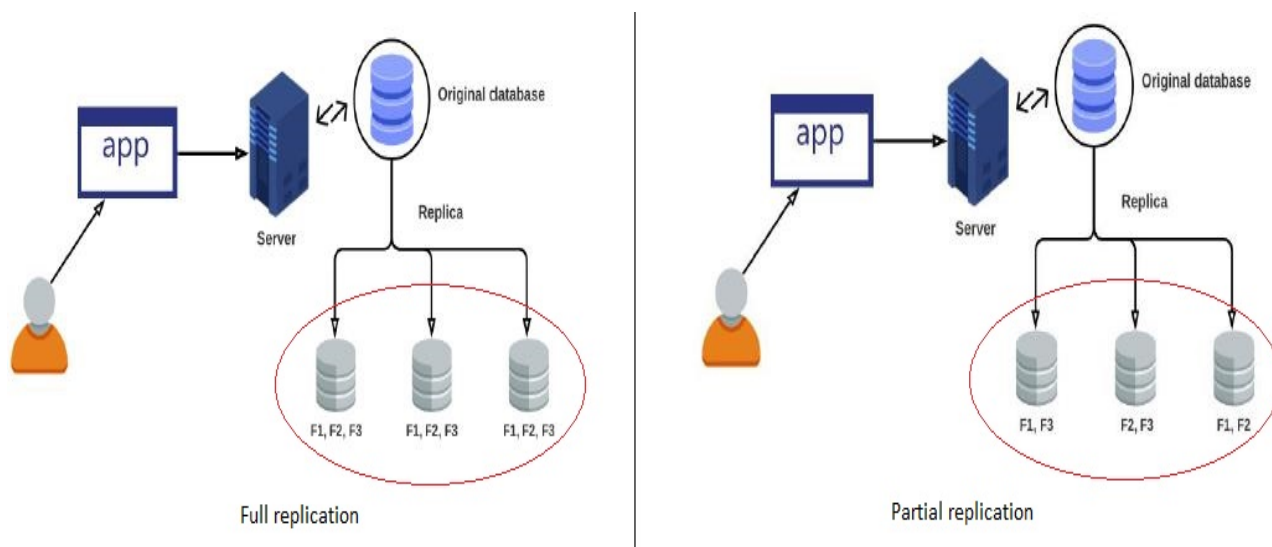
Χωρίς αμφιβολία ο κατακερματισμός, ανεξαρτήτως επιλογής αρχιτεκτονικής, σε ένα κατανεμημένο σύστημα διαχείρισης δεδομένων αυξάνει την πολυπλοκότητα του συστήματος αλλά σαφέστατα επηρεάζει θετικά την απόδοση του εξασφαλίζοντας ταυτόχρονα και ασφαλέστερη διαχείριση.



Εικόνα 10 - Παράδειγμα αρχιτεκτονικής υβριδικού κατακερματισμού με τη μορφή δέντρου κατάτμησης

### 4.3 Διαμερισμός των δεδομένων με αντίγραφα (*replication*)

Ο διαμερισμός με αντίγραφα (*replication*) συνιστά τη δεύτερη εναλλακτική πρόταση τμηματοποίησης των δεδομένων ενός κατακερμασμένου συστήματος. Είναι η διαδικασία κατά την οποία δημιουργούνται και διανέμονται αντίγραφα των δεδομένων που έχουν επιλεγεί, με τη δέσμευση ότι οποιαδήποτε τροποποίηση δεχτούν τα συγκεκριμένα δεδομένα οι αλλαγές αυτές θα πρέπει να είναι τηρούνται και στα κατάλληλα αντίγραφα. Σε μία κατακερμασμένη βάση μπορούν να αποθηκευτούν πολλά αντίγραφα κάθε σχέσης ή ένα τμήμα μια σχέσης σε παραπάνω από μία τοποθεσία. Όταν ολόκληρη σχέση αναπαράγεται σε μία ή παραπάνω τοποθεσίες έχουμε *πλήρης αντιγραφή* (*full replication*), ενώ *μερική αντιγραφή* (*partial replication*) όταν μόνο μερικά τμήματα της σχέσης αντιγράφονται, σύμφωνα με την αναγκαιότητα των δεδομένων κάθε τοποθεσίας (εικόνα 11). Ο διαμερισμός μπορεί να πραγματοποιηθεί μέσω *σύγχρονης* (*synchronous replication*) ή *ασύγχρονης* (*asynchronous replication*) αναπαραγωγής. Η βασική διαφορά τους έγκειται στον τρόπο με τον οποίο αποθηκεύονται/ενημερώνονται τα δεδομένα. Στη σύγχρονη αναπαραγωγή τα δεδομένα αντιγράφονται ταυτόχρονα στο βασικό χώρο αποθήκευσης και στα αντίγραφα, επομένως απαιτείται η σχέση και τα αντίγραφά της να είναι πάντα συγχρονισμένα. Αντιθέτως, στον ασύγχρονο διαμερισμό τα δεδομένα μεταδίδονται στα αντίγραφα αφού πρώτα έχουν εγγραφεί στον κύριο χώρο αποθήκευσης, δηλαδή η αντιγραφή πραγματοποιείται σε περιοδικές φάσεις (π.χ. κάθε 2 λεπτά). Κατά συνέπεια επιτρέπεται, για σύντομο χρονικό διάστημα, έως ότου πραγματοποιηθεί η ασύγχρονη αντιγραφή, διαφορετικά αντίγραφα της ίδιας σχέσης να εμπεριέχουν διαφορετικές τιμές.



Εικόνα 11 - Σχηματική λειτουργία πλήρη και μερικού διαχωρισμού δεδομένων

Διασώζοντας αντίγραφα με τη λειτουργία του διαμερισμού των δεδομένων, ένα κατακευμεμένο σύστημα μπορεί να επωφεληθεί από κίνητρα τα οποία συνοψίζονται ακολούθως.

- Αποτελεί μια δημοφιλής τεχνική ανοχής σφαλμάτων των κατακευμεμένων βάσεων καθώς εάν ένας κόμβος παρουσιάσει κάποια δυσλειτουργία η δραστηριότητα του μπορεί να συνεχιστεί χρησιμοποιώντας ένα αντίγραφο από έναν άλλο κόμβο, κάνοντας έτσι το σύστημα πιο αξιόπιστο. Η διάθεση αντιγράφων καθιστά το σύστημα περισσότερο διαθέσιμο, καθώς υπάρχει η δυνατότητα να χρησιμοποιηθεί ένας άλλος κόμβος χωρίς να υπάρξει μεγάλη διάρκεια διακοπής λειτουργίας. Επιπροσθέτως, το σύστημα δύναται να λειτουργήσει ακόμα και αν υπάρξει πρόβλημα στο δίκτυο επικοινωνίας.
- Αυξάνει την αποδοτικότητα, καθώς με τη χρήση πολλών αντιγράφων το σύστημα επεξεργάζεται περισσότερα δεδομένα σε λιγότερο χρόνο. Γίνεται πιο γρήγορο και αυτό διότι, πρώτον μειώνεται ο λανθάνων χρόνος (*latency*) των ερωτημάτων και δεύτερον γιατί διατηρούνται αντίγραφα δεδομένων γεωγραφικά πιο κοντά στον χρήστη μειώνοντας τον χρόνο επικοινωνίας.
- Ενισχύει την κλιμάκωση (*scalability*) του συστήματος. Η προσθήκη επιπλέον κόμβων και η δυνατότητα επιπρόσθετων αντιγράφων προσφέρει τη δυνατότητα επεξεργασίας ακόμα περισσότερων δεδομένων και εξυπηρέτησης περισσότερων ερωτημάτων. Όσο περισσότεροι οι κόμβοι και τα αντίγραφα, τόσο μειώνεται ο λανθάνων χρόνος (*latency*) των ερωτημάτων και αυξάνεται η ταχύτητα επεξεργασίας και απάντησης των ερωτημάτων.

Η διαδικασία διαμερισμού των αντιγράφων θα πρέπει να αντιμετωπίζεται ως ένα πρόβλημα βελτιστοποίησης τοποθέτησης κάθε αντιγράφου. Ο επιτυχής διαμερισμός μπορεί να οριστεί από δύο επιδιώξεις. Η πρώτη αφορά την ελαχιστοποίηση του κόστους, όπου το συνολικό ελάχιστο κόστος υπολογίζεται αθροίζοντας το κόστος επικοινωνίας (μετάδοση μηνυμάτων και σχετικά

δεδομένα), το κόστος ενοποίησης και ενημέρωσης των λειτουργιών κάθε τμήματος σε κάθε κόμβο (χρήση *CPU* και λειτουργία εισόδου/εξόδου των δεδομένων) και το κόστος αποθήκευσης κάθε τμήματος σε κάθε κόμβο. Η δεύτερη επιδίωξη αφορά στη μεγιστοποίηση της απόδοσης η οποία υπολογίζεται ειδικά από τους χρόνους απόκρισης του συστήματος κατά τη διάρκεια εγγραφής και ανάγνωσης δεδομένων. Ο χρόνος απόκρισης μίας συναλλαγής εκτιμάται προσθέτοντας το χρόνο μετάδοσης των δεδομένων και του χρόνου επεξεργασίας των λειτουργιών εγγραφής/ανάγνωσης των δεδομένων. Αποτελεσματικός, συνεπώς, θεωρείται ο διαμερισμός ο οποίος ισορροπεί μεταξύ του κόστους που επιφέρει η αποθήκευση (η διαβίβαση και η επεξεργασία των δεδομένων), ο περιορισμός διανομής των δεδομένων και η απόδοση, ιδιαίτερα στο χρόνο απόκρισης των ερωτημάτων. Οι προκλήσεις που θα πρέπει να αντιμετωπιστούν για την επίτευξη των παραπάνω στόχων του διαμερισμού των αντιγράφων είναι:

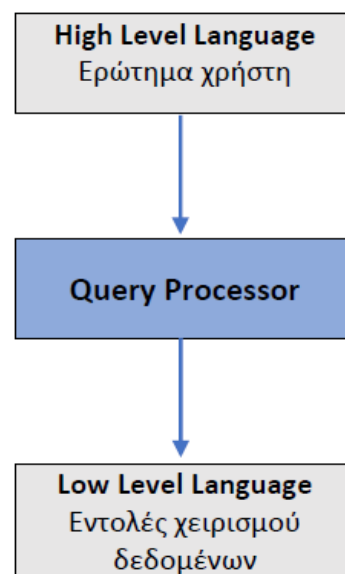
- ✓ η εφαρμογή των πολύπλοκων μηχανισμών της τοποθέτησης και αναζήτησης των αντιγράφων (η οποία σε ετερογενή κατανεμημένα συστήματα απαιτεί επιπλέον προσπάθεια),
- ✓ η διάδοση των δεδομένων μεταξύ των αντιγράφων (π.χ. η ενημέρωσή τους),
- ✓ η συντήρηση συνέπειας,
- ✓ η πολυπλοκότητα η οποία μερικές φορές είναι εμφανής στους χρήστες,
- ✓ το εύρος του δικτύου (*bandwidth*) που καταναλώνουν τα πρωτόκολλα και
- ✓ η αδυναμία να επιτευχθεί απόλυτη διαφάνεια (*transparency*) αναπαραγωγής στους χρήστες.

# 5

## Κόστος διαχείρισης των κατακευμαμένων δεδομένων

Ένα κατακευμαμένο σύστημα διαχείρισης δεδομένων θα πρέπει να είναι αποδοτικό και αυτό προκύπτει και από την αποτελεσματική απάντηση των ερωτημάτων (*queries*) που θέτουν οι χρήστες του συστήματος. Τα ερωτήματα εκφράζονται από τους χρήστες χρησιμοποιώντας μία υψηλού επιπέδου γλώσσα προγραμματισμού, όπως π.χ. η *SQL* (*Structure Query Language*), σε μία λογικά κατακευμαμένη βάση (έτσι όπως την αντιλαμβάνονται οι χρήστες), τα οποία ένας επεξεργαστής ερωτημάτων (*query processor*) αναλαμβάνει να τα μετατρέψει σε μία ισοδύναμη, χαμηλού επιπέδου γλώσσα, στις φυσικά κατακευμαμένες τοπικές βάσεις δεδομένων (εικόνα 12). Η μετατροπή οφείλει να επιτυγχάνει την ορθότητα και την αποτελεσματικότητα του ερωτήματος. Η μεταβολή αυτή θεωρείται ορθή μόνο όταν το ερώτημα χαμηλού επιπέδου παράγει το ίδιο αποτέλεσμα με το αρχικό ερώτημα. Ένα ερώτημα μπορεί να περιλαμβάνει ένα αίτημα για την παραγωγή αποτελεσμάτων δεδομένων από τη βάση δεδομένων ή για την πραγματοποίηση μίας ενέργειας στα δεδομένα (π.χ. τροποποίηση, διαγραφή) ή και τα δύο μαζί. Τα ερωτήματα προκύπτουν από τον σχεδιασμό αλγορίθμων, σκοπός των οποίων είναι να τα μετατρέψουν σε μία σειρά λειτουργιών χειρισμού των δεδομένων. Οι αλγόριθμοι θα πρέπει να σχεδιαστούν με τέτοιο τρόπο που όχι μόνο θα μπορούν να παράγουν σωστά αποτελέσματα αλλά και αποδοτικά, εννοώντας με το μικρότερο δυνατό κόστος. Ποιοι όμως είναι οι παράγοντες που καθορίζουν το κόστος ενός ερωτήματος; Το κόστος αναφέρεται στους παράγοντες που είναι υπεύθυνοι για την κατανάλωση των πόρων του συστήματος, καθώς και για την καθυστέρηση που προκύπτει από τη στιγμή που θα τεθεί ένα ερώτημα στη βάση, έως την οριστική προβολή του αποτελέσματος. Καθώς τα κατακευμαμένα συστήματα είναι συνήθως πολύπλοκα και πολλές φορές ανομοιογενή, η σχεδίαση ενός αποδοτικού ερωτήματος αποτελεί ένα δύσκολο εγχείρημα.

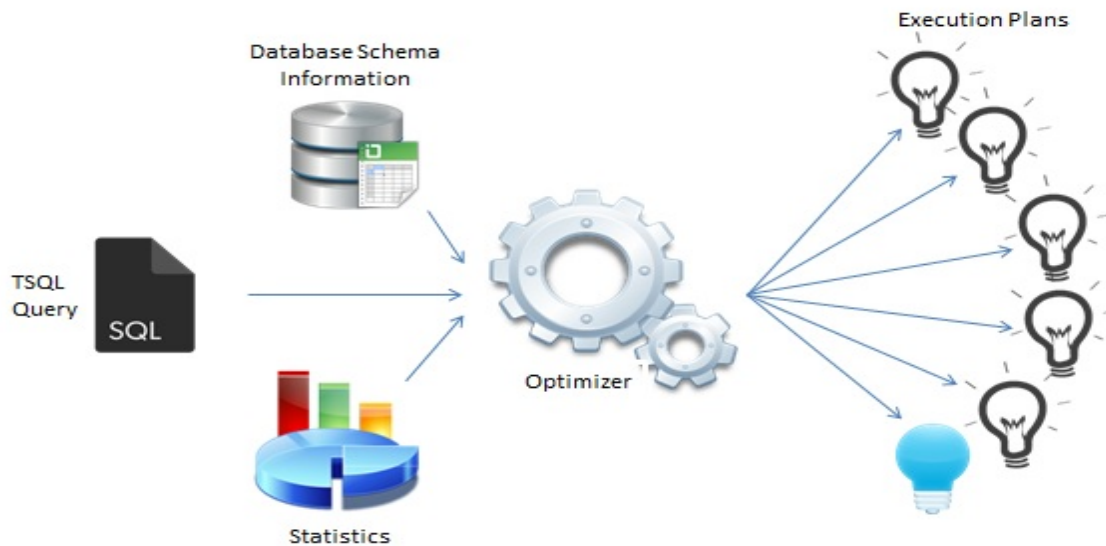
Ένα ερώτημα συνήθως έχει πολλές πιθανές στρατηγικές εκτέλεσης. Η επιλογή των πιο αποτελεσματικών ερωτημάτων εκτέλεσης μίας δήλωσης *SQL* αναφέρεται ως διαδικασία βελτιστοποίησης ερωτημάτων (*query optimization*). Τη διαδικασία αυτή την αναλαμβάνει ένα σημαντικό λογισμικό πρόγραμμα της βάσης δεδομένων, ο βελτιστοποιητής (*query optimizer*), ο



Εικόνα 12 - Λειτουργία επεξεργαστή ερωτημάτων



οποίος καθορίζει την αποδοτικότερη μέθοδο πρόσβασης στα δεδομένα για ένα ερώτημα *SQL* (εικόνα 13). Ο βελτιστοποιητής βασίζεται σε στατιστικά στοιχεία, για την επιλογή του αποδοτικότερου τρόπου εκτέλεσης του ερωτήματος, που χρησιμοποιούνται για την προσπέλαση των δεδομένων της κατανεμημένης βάσης. Συνεπώς, η εκτίμηση του κόστους είναι προσεγγιστική και όχι πραγματική.



Εικόνα 13 - Λειτουργία βελτιστοποιητή ερωτημάτων

Στα κεντροποιημένα συστήματα βάσεων δεδομένων εξετάζονται όλοι οι πιθανοί τρόποι εκτέλεσης ενός ερωτήματος και επιλέγεται ο οικονομικότερος. Το εκτιμώμενο κόστος βασίζεται κυρίως στο κόστος χρήσης του επεξεργαστή (*CPU*) και της εισόδου/εξόδου (*I/O*) των δεδομένων. Στα κατανεμημένα συστήματα βάσεων δεδομένων η βελτιστοποίηση ερωτήσεων περιλαμβάνει δύο μέρη: τη στρατηγική βελτιστοποίησης του ερωτήματος και τη βελτιστοποίηση της τοπικής επεξεργασίας του ερωτήματος. Η στρατηγική της βελτιστοποίησης του ερωτήματος είναι η πιο σημαντική από τα δύο. Πιθανότατα υπάρχουν αρκετοί διαφορετικοί τρόποι επίλυσης ενός ερωτήματος καθώς τα δεδομένα αποθηκεύονται σε διαφορετικές τοποθεσίες. Συνεπώς, οι πόροι του συστήματος και ο χρόνος απόκρισης κάθε στρατηγικής είναι εξίσου διαφορετικοί, ενώ, όπως αναμένεται, το συνολικό εκτιμώμενο κόστος θα πρέπει να περιλαμβάνει και το αντίστοιχο κόστος επικοινωνίας μεταξύ των κόμβων. Η εκτέλεση ενός ερωτήματος περιλαμβάνει μόνο την τοπική επεξεργασία (σε ένα κόμβο) αποκλειστικά στην περίπτωση που δεν έχουμε κατακερματίσει μία σχέση. Αντιθέτως, εάν υπάρχει κατακερματισμός μία σχέσης, θα πρέπει οπωσδήποτε να υπολογιστεί επιπροσθέτως και το κόστος επικοινωνίας μεταξύ των συσχετιζόμενων κόμβων. Η εκτέλεση της επιλεγμένης στρατηγικής πραγματοποιείται από το ερώτημα που έχει δηλωθεί από τη χαμηλού επιπέδου γλώσσα (*Low Level*). Καθώς τα δεδομένα είναι γεωγραφικά μοιρασμένα στην κατανεμημένη βάση δεδομένων η επεξεργασία ενός ερωτήματος συντίθεται μέσα από τρεις φάσεις:



- I. Τη φάση τοπικής επεξεργασίας, η οποία βασικά περιλαμβάνει την τοπική επεξεργασία των δεδομένων και με πιο γνωστές τις λειτουργίες:  
*select(σ)*: που επιστρέφει τις πλειάδες που ικανοποιούν μία συγκεκριμένη συνθήκη.  
*project(π)*: που επιστρέφει τα χαρακτηριστικά που αναφέρονται.
- II. Τη φάση μείωσης όπου χρησιμοποιείται μία ακολουθία από λειτουργίες που μειώνουν το μέγεθος των σχέσεων (π.χ. join και semijoin που επιστρέφουν επιλεγμένα δεδομένα από δύο ή περισσότερους πίνακες).
- III. Την τελική φάση επεξεργασίας όπου όλα τα αποτελέσματα των σχέσεων στέλνονται στη τοποθεσία συναρμολόγησης, εκεί δηλαδή όπου το τελικό αποτέλεσμα του ερωτήματος συντίθεται.

Στα ερωτήματα των καταναμημένων συστημάτων, ο τεμαχισμός μία σχέσης σε τμήματα, η ένωση των τμημάτων αυτών, για να σχηματιστεί πάλι ορθά μία ολόκληρη σχέση, καθώς και η μεταφορά των τμημάτων αυτών από τη μία βάση σε μία άλλη θεωρούνται πολύ κοινές λειτουργίες.

Όπως επισημάνθηκε και παραπάνω ο όρος *κόστος* αναφέρεται στους παράγοντες που είναι υπεύθυνοι για την κατανάλωση των πόρων του συστήματος καθώς και για την καθυστέρηση που προκύπτει από τη στιγμή που εκφραστεί ένα ερώτημα έως την οριστική προβολή του αποτελέσματος. Τα συστατικά που περιλαμβάνει το εκτιμώμενο συνολικό κόστος εκτέλεσης ενός ερωτήματος είναι:

- Το κόστος πρόσβασης στα δευτερεύοντα αποθηκευτικά μέσα (*access cost to secondary storage*). Ο χρόνος που απαιτείται για τη φόρτωση των δεδομένων από τη δευτερεύουσα αποθηκευτική συσκευή στην κύρια μνήμη. Ο χρόνος αυτός επηρεάζεται από το μέγεθος των δεδομένων που θα πρέπει να ανακτηθούν, το μέγεθος του διαθέσιμου *buffer* καθώς και τη ταχύτητα των χρησιμοποιημένων συσκευών.
- Το κόστος αποθήκευσης των δεδομένων (*storage cost*), του χώρου δηλαδή που θα χρησιμοποιηθεί στα δευτερεύοντα αποθηκευτικά μέσα, καθώς και της μνήμης για την εκτέλεση του ερωτήματος.
- Το κόστος υπολογισμού (*computation cost*), δηλαδή ο χρόνος που απαιτείται από τον επεξεργαστή (*CPU*) για την ολοκλήρωση εκτέλεσης των λειτουργιών του ερωτήματος. Η λειτουργία αυτή αναφέρεται στην αναζήτηση, ταξινόμηση, εγγραφή, συγχώνευση και υπολογισμό των πεδίων τιμών.
- Το κόστος χρήσης της κύριας μνήμης (*memory uses cost*), που σχετίζεται με τον αριθμό των *buffers* που απαιτούνται για την εκτέλεση του ερωτήματος.
- Το κόστος επικοινωνίας (*communication cost*), το οποίο αφορά στο χρόνο της καταναμημένης επικοινωνίας, αποστολής και λήψης τμημάτων της καταναμημένης βάσης που βρίσκονται σε πολλούς διακομιστές ή κόμβους. Αποτελεί το σημαντικότερο παράγοντα υπολογισμού του συνολικού κόστους του ερωτήματος σε ένα καταναμημένο σύστημα και διαμορφώνεται από το άθροισμα της αρχικοποίησης του ερωτήματος και της μετάδοσης των δεδομένων.

Σε μία καταναμημένη βάση δεδομένων, κατά την επεξεργασία των ερωτημάτων, εκτός από την αποτελεσματικότητα των ερωτημάτων θα πρέπει να εξασφαλίζεται η διαθεσιμότητα των πόρων

καθώς και η αξιοπιστία τους. Σε ένα κατανεμημένο περιβάλλον εξετάζονται δύο εκτιμήσεις κατανάλωσης χρόνου. Η πρώτη αφορά στο χρόνο τοπικής επεξεργασίας (*Local Processing time*) που παράγεται από το άθροισμα του κόστους λειτουργιών των δεδομένων που συντελούνται στην κεντρική μνήμη των συμμετεχόντων τοποθεσιών (*CPU cost*,) και του κόστους των ενεργειών αποθήκευσης των δεδομένων στα δευτερεύοντα αποθηκευτικά αντίστοιχα (*I/O cost*), δηλαδή:

$$\text{Local Processing time} = \text{CPU cost} + \text{I/O cost}.$$

Η δεύτερη εκτίμηση που εξετάζεται σχετίζεται με το κόστος επικοινωνίας (*communication cost*) που παράγεται από την ανταλλαγή δεδομένων μεταξύ των συμμετεχόντων τοποθεσιών. Το κόστος επικοινωνίας εξαρτάται από επιμέρους παράγοντες όπως το μέγεθος των μεταφερόμενων δεδομένων, την επιλογή της καταλληλότερης τοποθεσίας για την εκτέλεση του ερωτήματος, τον αριθμό των μηνυμάτων που θα μεταφερθούν μεταξύ των συσχετιζόμενων τοποθεσιών και φυσικά το δίκτυο επικοινωνίας. Στην περίπτωση, όμως, που μία σχέση δεν έχει κατακερματιστεί τότε δεν περιλαμβάνεται το κόστος επικοινωνίας.

Συνεπώς η εκτίμηση του **συνολικού κόστους** ενός ερωτήματος σε ένα κατανεμημένο σύστημα ορίζεται ως εξής:

$$\text{Total cost} = \text{CPU cost} + \text{I/O cost} + \text{Communication cost}$$

Πιο αναλυτικά:  $\text{CPU cost} = \text{unit instruction cost} * \text{no. of instructions}$

$$\text{I/O cost} = \text{unit disk I/O cost} * \text{no. of disk I/O's}$$

$$\text{Communication cost} = \text{message initiation} + \text{transmission}$$

Όπου:  $\text{unit instruction cost}$  = κόστος εκτέλεσης μίας εντολής από τον επεξεργαστή.

$\text{No. of instructions}$  = ο αριθμός των εντολών προς εκτέλεση.

$\text{I/O cost}$  = το κόστος αποθήκευσης.

$\text{unit disk I/O}$  = χώρος αποθήκευσης σε ένα αποθηκευτικό μέσο.

$\text{message initiation}$  = αρχικοποίηση του μηνύματος που είναι προς εκτέλεση.

$\text{transmission}$  = η μετάδοση του μηνύματος.

# 6

## Αρχιτεκτονική μοντέλων παράλληλων και κατανεμημένων συστημάτων

Η αρχιτεκτονική ενός παράλληλου ή/και κατανεμημένου συστήματος καθορίζει και τη δομή του. Δηλαδή τα συστατικά (clients, servers, network) που προσδιορίζουν το σύστημα, τη λειτουργία κάθε συστατικού, η οποία καθορίζεται συγκεκριμένα, ενώ ορίζονται και οι αλληλεπιδράσεις μεταξύ αυτών των συστατικών. Ο προσδιορισμός της αρχιτεκτονικής απαιτεί τον εντοπισμό των διαφόρων συστατικών του συστήματος με τις διεπαφές του και τις σχέσεις του αναφορικά με τα δεδομένα, καθώς και με τον έλεγχο τους μέσα από το σύστημα. Δηλαδή, η τοποθέτηση των συστατικών του κατανεμημένου συστήματος, η οργάνωση τους και ο καθορισμός των σχέσεων τους αποτελούν τους βασικούς άξονες της αρχιτεκτονικής των συστημάτων.

### 6.1 Κατανεμημένα συστήματα

Δύο είναι οι βασικές αρχιτεκτονικές ανάπτυξης κατανεμημένων συστημάτων.

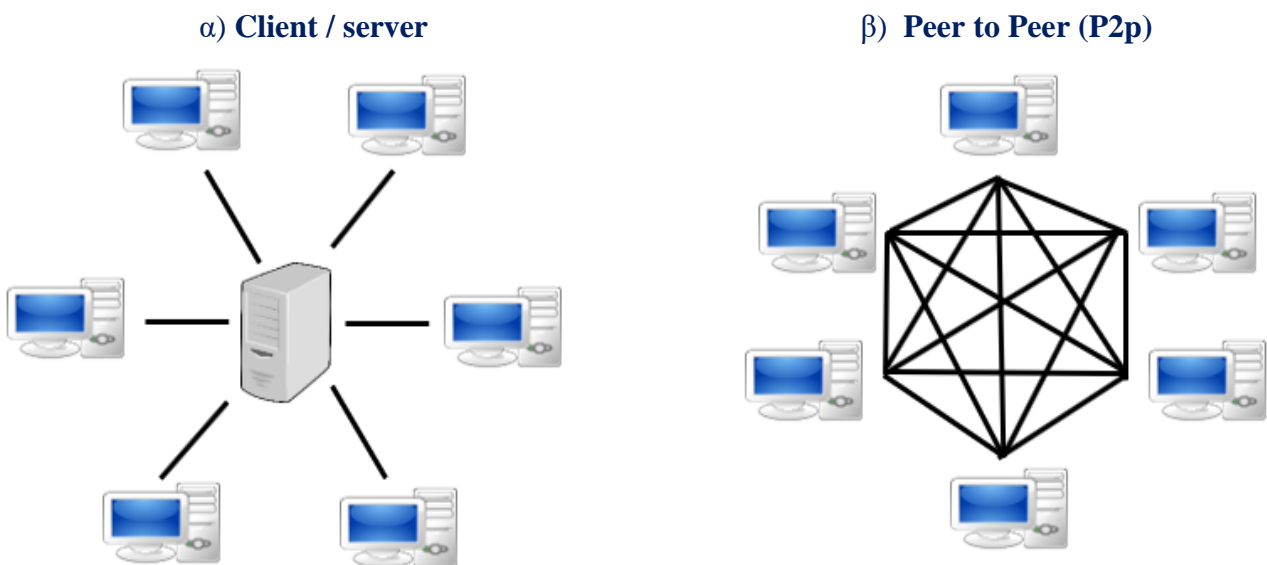
#### 1. Αρχιτεκτονική πελάτη/εξυπηρετητή (client/server)

Αποτελεί τη δημοφιλέστερη, επί του παρόντος, αρχιτεκτονική. Στο μοντέλο αυτό μία ομάδα από υπολογιστές πελάτες (clients) έχουν πρόσβαση σε μία βάση δεδομένων ενός εξυπηρετητή (ή διακομιστή-server) και αποστέλλουν αιτήματα για παροχή υπηρεσιών μέσω εφαρμογών. Ο εξυπηρετητής λαμβάνει το αίτημα και επιστρέφει την απάντηση ως ανταπόκριση στο αίτημα του πελάτη. Πελάτες και εξυπηρετητής τυπικά είναι διαφορετικοί υπολογιστές οι οποίοι συνδέονται μεταξύ τους μέσω ενός δικτύου υπολογιστών (εικόνα 14α). Ο πελάτης στέλνει το αίτημα μέσω εφαρμογής η οποία είναι μία διαδικασία ή ένα πρόγραμμα που διαβιβάζει μηνύματα σε έναν εξυπηρετητή μέσω δικτύου. Τα μηνύματα αυτά ζητούν από τον εξυπηρετητή την εκτέλεση μίας συγκεκριμένης εργασίας, π.χ. την αναζήτηση ενός αρχείου σε μια βάση δεδομένων ή την εγγραφή ενός αρχείου στο δίσκο του εξυπηρετητή. Στη συνέχεια το λογισμικό του εξυπηρετητή λαμβάνει τα μηνύματα αυτά, τα οποία μεταδίδονται μέσω του δικτύου, και εκτελεί τις αντίστοιχες ενέργειες (π.χ. ανάγνωση αρχείων της βάσης ή εκτέλεση ερωτημάτων) με σκοπό την ικανοποίηση των αιτημάτων αυτών. Οι εξυπηρετητές τυπικά τρέχουν πάνω σε πολύ ισχυρούς υπολογιστές για να μπορέσουν να ανταπεξέλθουν αποδοτικά στα αιτήματα των πελατών. Μια ακόμη πιο ευέλικτη κατανεμημένη αρχιτεκτονική αποτελεί το μοντέλο των πολλαπλών-πελατών/πολλαπλών-

εξυπηρετητών όπου η βάση δεδομένων κατανέμεται σε πολλαπλούς εξυπηρετητές οι οποίοι οφείλουν να επικοινωνούν μεταξύ τους σε κάθε ερώτημα αλλά και σε κάθε εκτέλεση συναλλαγών των πελατών. Κάθε πελάτης δημιουργεί μία σύνδεση με έναν εξυπηρετητή ο οποίος κατευθύνει τα αιτήματα του πελάτη. Τα δεδομένα μπορούν να σταλούν από τους εξυπηρετητές σε πολλούς πελάτες μέσα στο δίκτυο, οι οποίοι στη συνέχεια τα συναρμολογούν ή τα επεξεργάζονται, ενώ η επικοινωνία μεταξύ των εξυπηρετητών συντελείται με τρόπο απόλυτα διαφανή για τους χρήστες.

## 2. Αρχιτεκτονική ομοτίμων (*peer-to-peer / P2P*)

Το μοντέλο των ομοτίμων (P2P) αποτελεί ένα πραγματικά καταναμημένο σύστημα διαχείρισης δεδομένων καθώς δεν υπάρχει διάκριση μεταξύ ενός πελάτη και ενός εξυπηρετητή. Το μοντέλο P2P αποτελείται από ένα αποκεντρωμένο δίκτυο από τερματικά συστήματα (peers) τα οποία επικοινωνούν απευθείας μεταξύ τους χωρίς να υπάρχουν διακρίσεις. Κάθε κόμβος είναι και πελάτης και εξυπηρετητής (εικόνα 14β). Ο φόρτος εργασίας διανέμεται μεταξύ των ομοτίμων, όλοι συνεισφέρουν και καταναλώνουν πόρους εντός του δικτύου χωρίς την ανάγκη ύπαρξης ενός κεντρικού εξυπηρετητή. Η αρχιτεκτονική P2P χαρακτηρίζεται από την πολυπλοκότητα διαχείρισης των καταναμημένων δεδομένων της και τη δυσκολία στην υλοποίησή της.



Εικόνα 14 - Γενικά μοντέλα αρχιτεκτονικής παράλληλων και καταναμημένων συστημάτων

## 6.2 Παράλληλα συστήματα

Κι εδώ, όπως και στα καταναμημένα συστήματα δύο είναι οι κυρίαρχες προσεγγίσεις των μοντέλων παράλληλης αρχιτεκτονικής.

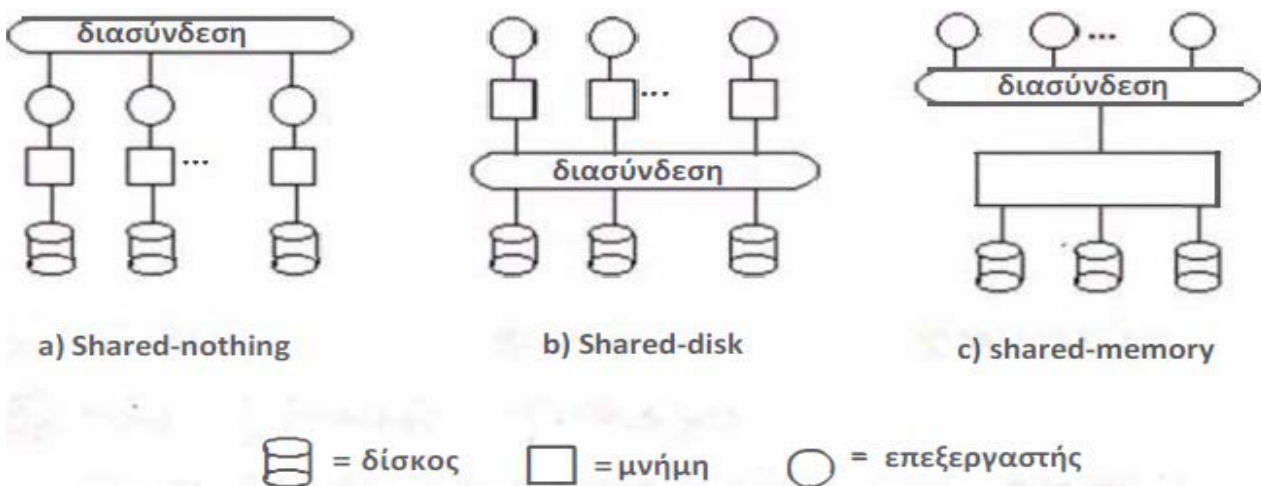
### 1. Αρχιτεκτονική Shared-nothing

Κάθε επεξεργαστής λειτουργεί ανεξάρτητα, έχει αποκλειστική πρόσβαση στην μνήμη του και στο αποθηκευτικό του μέσο. Κάθε κόμβος προβάλλεται ως μια τοπική τοποθεσία, με τη δική του βάση δεδομένων και το λογισμικό, σε μια καταναμημένη βάση δεδομένων. Η διαφορά μίας shared-nothing παράλληλης βάσης δεδομένων και μίας καταναμημένης βάσης δεδομένων έγκειται στον τρόπο υλοποίησης της πλατφόρμας. Συνεπώς, οι περισσότερες λύσεις που έχουν σχεδιαστεί για καταναμημένες βάσεις δεδομένων μπορούν να επαναχρησιμοποιηθούν και σε παράλληλα συστήματα διαχείρισης βάσεων δεδομένων. Βασικό στόχο της υλοποίησης της συγκεκριμένης αρχιτεκτονικής αποτελεί η αποφυγή της διαμάχης μεταξύ των συνδεδεμένων κόμβων στη διαμοίραση των πόρων του δικτύου τους.

### 2. Αρχιτεκτονική Shared-memory/Shared-disk

Στην αρχιτεκτονική διαμερισμού των πόρων κάθε κόμβος του δικτύου διαμοιράζεται ένα κομμάτι των πόρων του δικτύου. Στο μοντέλο shared-memory τις μονάδες μνήμης, αλλά διατηρεί και διαχειρίζεται το δικό του αποθηκευτικό μέσο, ενώ αντίστοιχα στο μοντέλο shared-disk διαμοιράζονται τα αποθηκευτικά μέσα του δικτύου και ο κάθε κόμβος έχει την δική του μνήμη (εικόνα 15). Με την υλοποίηση του συγκεκριμένου αρχιτεκτονικού μοντέλου επιτυγχάνεται ταχεία προσαρμοστικότητα σε μεταβαλλόμενο φόρτο εργασίας και απεριόριστη κλιμάκωση, που είναι και το μεγαλύτερο πλεονέκτημα της.

Συμπερασματικά, θα επισημαίναμε ότι το μοντέλο αρχιτεκτονικής shared-nothing επικεντρώνεται στη μεγιστοποίηση απόδοσης του παράλληλου συστήματος διαχείρισης των δεδομένων, ενώ το μοντέλο share memory/shared disk στη μεγιστοποίηση της χρήσης των διαθέσιμων πόρων του δικτύου.



Εικόνα 15 - Σχηματική απεικόνιση αρχιτεκτονικής παράλληλων συστημάτων (λογικό μοντέλο)

# 7

## *Εφαρμογή παράλληλων και κατανεμημένων αλγορίθμων σε περιβάλλον προσομοίωσης*

### *7.1 Χρήση αρχιτεκτονικού μοντέλου για τη μελέτη των παράλληλων Αλγορίθμων*

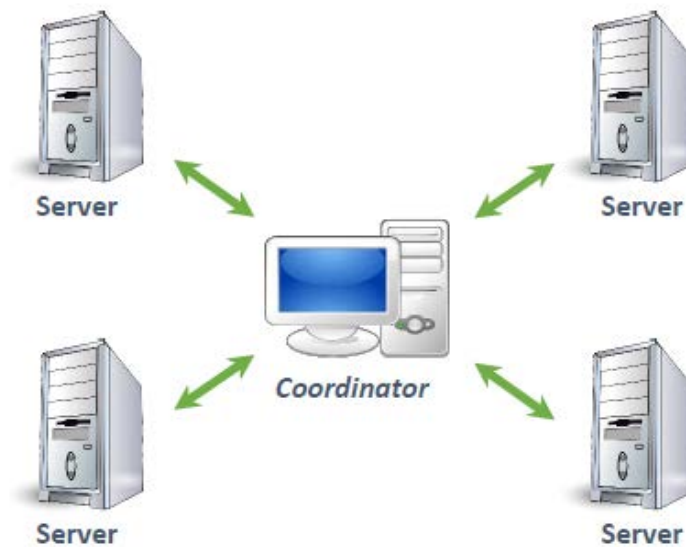
Βασικό σκοπό της ενότητας αποτελεί η μελέτη και η παρουσίαση των σημαντικότερων μηχανισμών λειτουργίας των αλγορίθμων που χρησιμοποιούνται για τη διαχείριση ενός παράλληλου και κατανεμημένου συστήματος δεδομένων. Για να γίνει πιο κατανοητός ο τρόπος λειτουργίας των αλγορίθμων θα χρησιμοποιηθεί ένα εικονικό σύστημα κατανεμημένης βάσης δεδομένων. Αυτό θα περιλαμβάνει:

- α) σε επίπεδο hardware, ένα κατανεμημένο σύστημα όμοιο με ένα από τα αρχιτεκτονικά μοντέλα που παρουσιάσαμε παραπάνω,
- β) μία μικρή, σε μέγεθος δεδομένων, κατανεμημένη βάση που αποτελείται από έναν δισδιάστατο πίνακα δεδομένων και που θα χρησιμοποιηθεί για τη μελέτη της λειτουργίας των αλγορίθμων, καθώς και για τον υπολογισμό του κόστους κάθε φάσης επεξεργασίας.

Ξεκινώντας, θα πρέπει, για αρχή, να επιλέξουμε ένα απλό, γενικό, αφηρημένο αρχιτεκτονικό μοντέλο το οποίο θα χρησιμοποιηθεί ως περιβάλλον προσομοίωσης. Έπειτα, θα δημιουργήσουμε μια εικονική κατανεμημένη βάση δεδομένων. Στη συνέχεια, θα αναλύσουμε, χρησιμοποιώντας βασικούς αλγορίθμους ερωτημάτων, βήμα προς βήμα, τον τρόπο λειτουργίας του κατανεμημένου συστήματος, τη διαδικασία υλοποίησης των αλγορίθμων, καθώς και το εκτιμώμενο κατά προσέγγιση κόστος κάθε φάσης επεξεργασίας, από τη στιγμή που ο χρήστης θέσει ένα ερώτημα έως την τελική λήψη της απάντησης.

Ως καταλληλότερο μοντέλο έχει επιλεγεί η αρχιτεκτονική shared-nothing προς χάριν της απλότητας αλλά και της ξεκάθαρης λειτουργίας κάθε φάσης επεξεργασίας των δεδομένων που επιτρέπουν να γίνουν κατανοητές οι διεργασίες, από την έναρξη υποβολής ενός ερωτήματος έως την προβολή του τελικού αποτελέσματος. Στην αρχιτεκτονική shared-nothing κάθε διακομιστής του δικτύου λειτουργεί ανεξάρτητα έχοντας αποκλειστική πρόσβαση στη μνήμη του και το αποθηκευτικό του μέσο. Τα δεδομένα που θα διαχειρίζεται το σύστημα που σχεδιάστηκε είναι κατανεμημένα στους τέσσερις διακομιστές του δικτύου και κάθε ένας διαχειρίζεται ένα τμήμα αυτών. Ένας επιπλέον διακομιστής θα αναλάβει το ρόλο του συντονιστή (coordinator) των ενεργειών που θα πρέπει να πραγματοποιούν οι υπόλοιποι διακομιστές για την εκτέλεση και

διεκπεραίωση των ερωτημάτων του χρήστη. Τα δεδομένα που έχει να διαχειριστεί το σύστημα συνθέτουν έναν πίνακα μιας βάσης δεδομένων ο οποίος είναι διαχωρισμένος ισομερώς βάσει οριζώντιου κατακερματισμού (*horizontal fragmentation*) στους τέσσερις διακομιστές (εικόνα 16). Κάθε ένας από αυτούς μπορεί να απευθύνει ένα ερώτημα στο συντονιστή, ο οποίος στη συνέχεια διανέμει το ερώτημα αυτό σε όλους τους συμμετέχοντες διακομιστές του δικτύου. Το ερώτημα αυτό ο συντονιστής μπορεί να το στείλει ακριβώς όπως το έλαβε ή και να το μετασχημασματοίσει με σκοπό την επίτευξη αποδοτικότερης επεξεργασίας από τους διακομιστές. Στη συνέχεια, οι διακομιστές επεξεργάζονται ταυτόχρονα το ερώτημα σε τοπικό επίπεδο. Όποιος ολοκληρώνει την επεξεργασία στέλνει πίσω στον συντονιστή το αποτέλεσμα του ερωτήματος, ο οποίος ακολούθως αναλαμβάνει να συνθέσει την τελική απάντηση και να τη στείλει στο χρήστη που έθεσε το ερώτημα. Για να επιστραφεί το τελικό αποτέλεσμα στο χρήστη θα πρέπει όλοι οι συμμετέχοντες διακομιστές να έχουν απαντήσει στο ερώτημα του συντονιστή. Εξαιρέση αποτελεί η περίπτωση κατά την οποία, βάσει της φύσης του ερωτήματος, ο συντονιστής δεν χρειάζεται να προβεί σε κάποια επιπλέον ενέργεια επεξεργασίας ή συγχώνευσης των απαντήσεων που έλαβε, όποτε και αποστέλλει διαδοχικά με σειρά κάθε απάντηση, έτσι όπως τη λαμβάνει από τους διακομιστές, άμεσα, χωρίς να αναμένει την απάντηση των υπολοίπων.



Εικόνα 16 - Αρχιτεκτονική Client /Server (shared-nothing)

Συνεπώς, η αποτελεσματικότητα της παράλληλης και καταναμημένης επεξεργασίας στο παραπάνω καταναμημένο μοντέλο, όπως άλλωστε συμβαίνει και σε όλα τα καταναμημένα συστήματα, εξαρτάται κυρίως από τον τρόπο με τον οποίο υλοποιούνται οι τρεις ακόλουθες βασικές φάσεις:

- I. Η τοπική επεξεργασία (*local processing*), δηλαδή ο χρόνος και οι πόροι που θα καταναλώσει κάθε διακομιστής για να επεξεργαστεί την ερώτηση που έστειλε ο συντονιστής. Θα πρέπει να πραγματοποιήσει μία απλή ή μια πιο σύνθετη επεξεργασία; Ο πιο αργός διακομιστής είναι αυτός που καθορίζει τον χρόνο αναμονής καθώς ο συντονιστής θα πρέπει να αναμένει,

στις περισσότερες των περιπτώσεων, την απάντηση όλων των διακομιστών για να συνθέσει την τελική απάντηση. Ως εκ τούτου, θα πρέπει όλοι οι διακομιστές να διαχειρίζονται, όσον είναι δυνατόν, τον ίδιο όγκο δεδομένων καθώς και να διαμοιράζονται το ίδιο φόρτο εργασίας.

- II. Το μέγεθος των μεταφερόμενων δεδομένων των τοπικών αποτελεσμάτων (*communication*). Μόλις κάθε διακομιστής ολοκληρώσει την επεξεργασία του ερωτήματος μεταδίδει το αποτέλεσμά του στο συντονιστή. Για την αποδοτικότερη επεξεργασία του ερωτήματος θα πρέπει ο όγκος των μεταφερόμενων αυτών δεδομένων να είναι συγκρίσιμος με τον όγκο του αποτελέσματος και όχι με τα αρχικά δεδομένα. Όσο μικρότερος ο όγκος των μεταφερόμενων δεδομένων τόσο αποδοτικότερος ο αλγόριθμος.
- III. Η αποτελεσματικότητα της φάσης συγχώνευσης των δεδομένων (*merging*). Όταν ο συντονιστής λάβει τα αποτελέσματα όλων των διακομιστών αναλαμβάνει να συνθέσει το τελικό αποτέλεσμα που θα στείλει στο χρήστη. Όσο μικρότερος ο όγκος των δεδομένων που δέχεται από τους διακομιστές για επεξεργασία, τόσο πιο σύντομα θα ολοκληρώσει τη φάση συγχώνευσης των δεδομένων και θα αποστείλει το τελικό αποτέλεσμα.

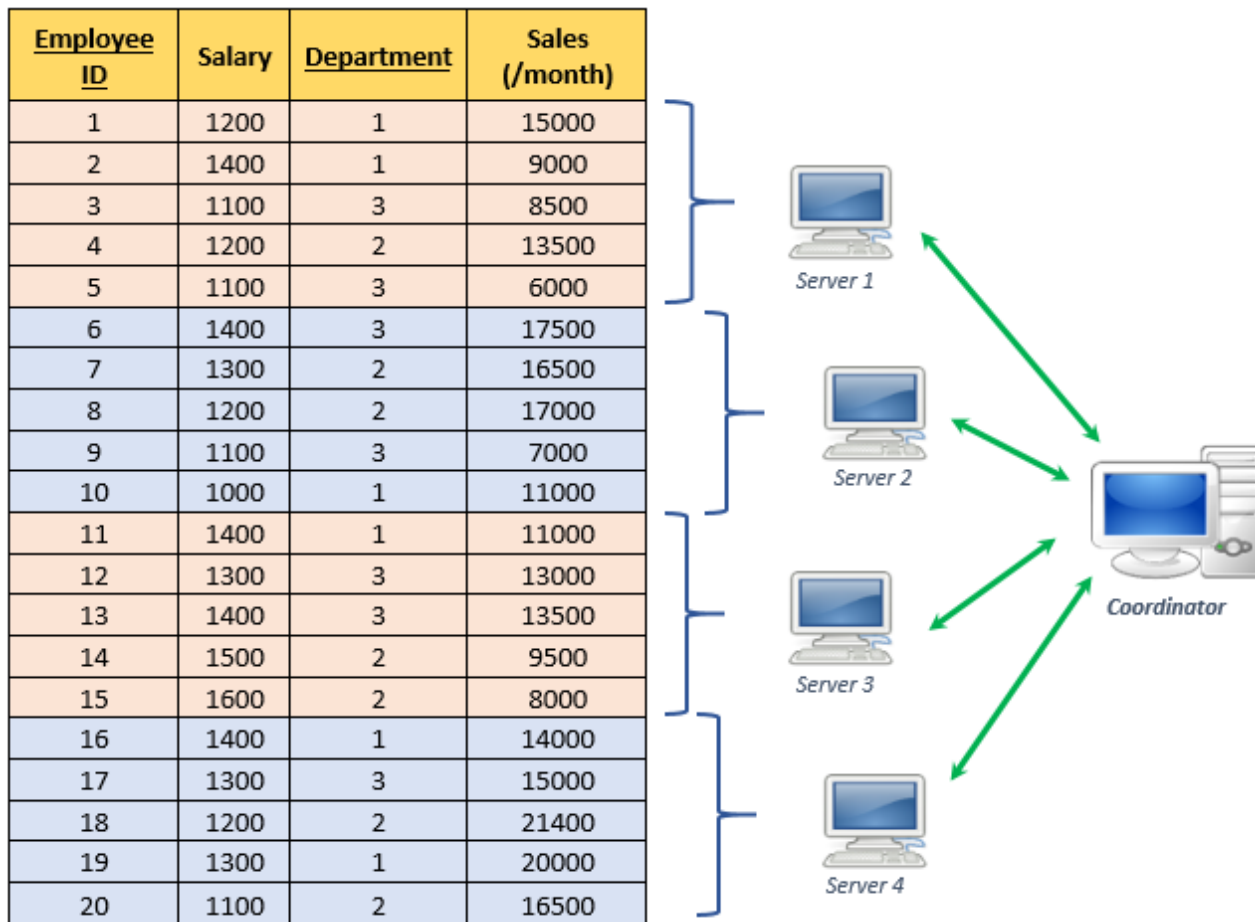
## 7.2 Υλοποίηση αλγορίθμων

Για να γίνει όσον το δυνατόν πιο ρεαλιστική η λειτουργία των αλγορίθμων καθώς και του κατανεμημένου συστήματος συνολικά, θα δημιουργήσουμε και θα χρησιμοποιήσουμε τυχαία στοιχεία των υπαλλήλων μίας εικονικής εμπορικής επιχείρησης που δραστηριοποιείται στον τομέα των πωλήσεων. Τα στοιχεία που θα έχουμε διαθέσιμα βρίσκονται σε μια κατανεμημένη βάση δεδομένων και αφορούν στο τμήμα των πωλήσεων της εταιρείας. Στη βάση αυτή καταχωρούνται ο μοναδικός κωδικός κάθε υπαλλήλου (*EmployeeID* – στην επιχείρηση εργάζονται 20 υπάλληλοι), ο μηνιαίος μισθός του (*Salary*), το υποκατάστημα στο οποίο εργάζεται (*Department* – η επιχείρηση διαθέτει τρία υποκαταστήματα), καθώς και το σύνολο των πωλήσεων (*Sales/month*) που πραγματοποιεί κάθε υπάλληλος ανά μήνα.

Τα δεδομένα, όπως αναφέρθηκε παραπάνω, έχουν διαχωριστεί και κατανεμηθεί ομοιόμορφα βάσει οριζόντιου κατακερματισμού στους τέσσερις διακομιστές της επιχείρησης οι οποίοι βρίσκονται σε διαφορετικές τοποθεσίες, ενώ τον έλεγχο της διαχείρισης των κατανεμημένων δεδομένων έχει αναλάβει ένας συντονιστής διακομιστής σύμφωνα με το ακόλουθο σχήμα.



## Κατανομή δεδομένων και σχεδιασμός αρχιτεκτονικής της Εμπορικής εταιρείας



Σχήμα 1 - Κατανομή των δεδομένων στο δίκτυο υπολογιστών της εμπορικής εταιρείας.

Βασική επιδίωξη αποτελεί η ανάθεση στο σύστημα τεσσάρων βασικών ερωτημάτων τα οποία θα μας επιτρέψουν να μελετήσουμε τον τρόπο λειτουργίας των αλγορίθμων. Τα ερωτήματα αυτά είναι βασικά, γενικά χρησιμοποιούμενα ερωτήματα επίλυσης προβλημάτων που χρησιμοποιούνται ευρέως σε κατακεντρωμένα συστήματα διαχείρισης δεδομένων και είναι τα εξής:

- 1) επιλογή εμφάνισης των δεδομένων που έχουν μία συγκεκριμένη τιμή,
- 2) υπολογισμός πεδίου τιμών βάσει κριτηρίων που δίνονται από τον χρήστη,
- 3) αναζήτηση της καλύτερης τιμής από ένα σύνολο δεδομένων και
- 4) ταξινόμηση αποτελεσμάτων βάσει συγκεκριμένης τιμής τους.

Για κάθε ερώτημα θα παρουσιαστούν αναλυτικά κάθε ένα από τα 3 βήματα (τοπική επεξεργασία των δεδομένων, μεταφορά των δεδομένων, τελική συγχώνευση των δεδομένων) των φάσεων επεξεργασίας των κατακεντρωμένων δεδομένων, από τη στιγμή που ο συντονιστής λάβει το ερώτημα έως και την τελική απάντησή του. Να σημειωθεί ότι για τα ερωτήματα 2, 3 και 4 θα παρουσιαστούν δύο εναλλακτικοί τρόποι υλοποίησης κάθε ερωτήματος για να διαπιστωθούν οι διαφορές στο τρόπο και στην αποδοτικότητα της λειτουργίας κάθε αλγορίθμου. Σκοπό της υλοποίησης των παραπάνω

ερωτημάτων αποτελεί η κατανόηση του τρόπου λειτουργίας των αλγορίθμων μελετώντας παράλληλα και τις αλλαγές που επιφέρει στο κόστος κάθε αλλαγή τροποποίησης της λειτουργίας τους, σύμφωνα πάντα με το ερώτημα που καλείται να διαχειριστεί το σύστημα. Στο τέλος κάθε ερωτήματος θα παρουσιαστούν τα αποτελέσματα της εκτίμησης, κατά προσέγγιση, του κόστους υλοποίησης κάθε αλγορίθμου.

### 7.3 Ερώτημα 1<sup>ο</sup> - Επιλογή τιμής

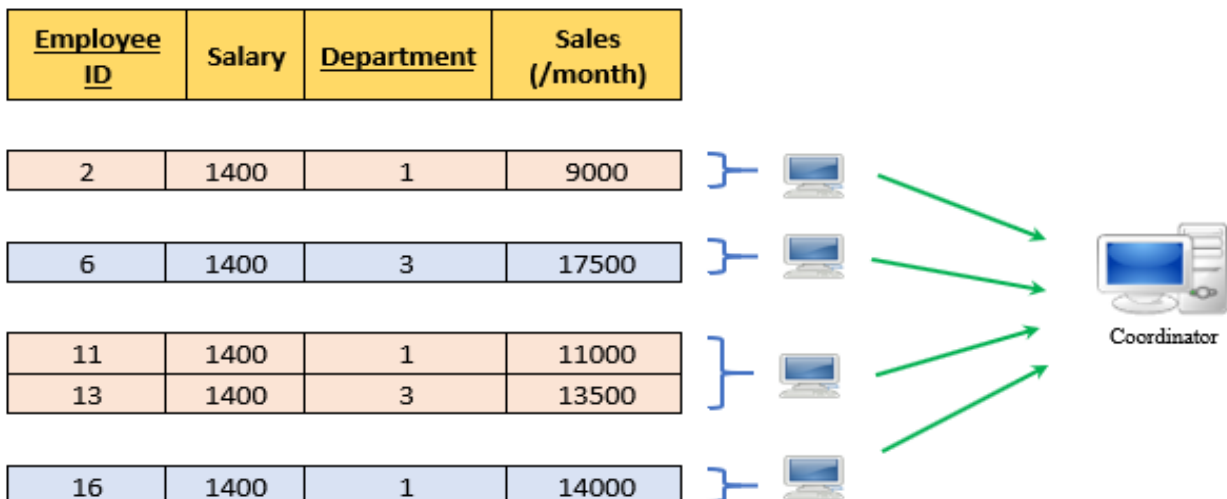
Η πιο απλή ερώτηση που μπορεί να θέσει ένας χρήστης σε ένα καταναμημένο σύστημα βάσης δεδομένων είναι της επιλογής εμφάνισης των δεδομένων που έχουν μια συγκεκριμένη τιμή.

Έστω ότι ένα χρήστης θέλει να δει το μισθό όλων των πωλητών της εταιρείας που είναι ίσος με 1.400 ( $Salary=1400$ ).

- Ο συντονιστής λαμβάνει το ερώτημα του χρήστη και στη συνέχεια το αποστέλλει σε όλους τους διακομιστές του συστήματος.

#### I. Τοπική επεξεργασία (Local Processing)

Για να απαντήσει το ερώτημα κάθε διακομιστής το μόνο που έχει να κάνει είναι να απορρίψει τις πλειάδες των οποίων η τιμή  $Salary \neq 1400$  και να αποστείλει στον συντονιστή όσες από αυτές πληρούν τη συνθήκη.



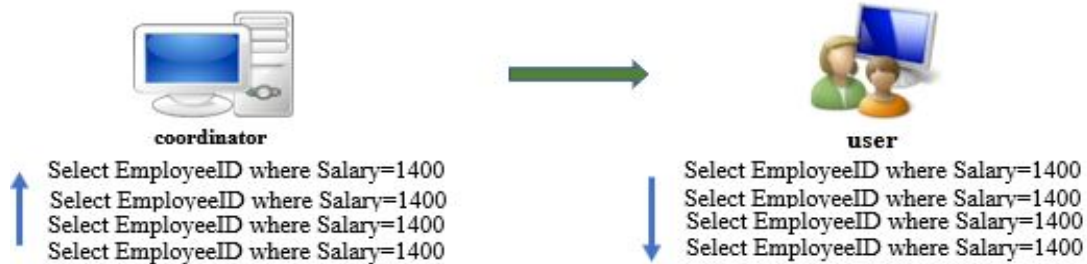
Σχήμα 2 - Φάση τοπικής επεξεργασίας των δεδομένων (Ερώτημα 1<sup>ο</sup>)

#### II. Μεταφορά δεδομένων (Communication)

Κάθε διακομιστής επιστρέφει στο συντονιστή τα δεδομένα των οποίων η τιμή  $Salary=1400$

### III. Συγχώνευση (Merging)

Ο συντονιστής στη συνέχεια αποστέλλει άμεσα τα αποτελέσματα κάθε διακομιστή στο χρήστη χωρίς καμία περαιτέρω επεξεργασία. Δεν χρειάζεται να περιμένει τα αποτελέσματα από όλους τους διακομιστές. Ό,τι πληροφορία λαμβάνει την αποστέλλει άμεσα χωρίς καμία καθυστέρηση κατά τη διαδικασία της συγχώνευσης.



Σχήμα 3 - Φάση συγχώνευσης δεδομένων (Ερώτημα 1<sup>ο</sup>)

#### **Εκτίμηση συνολικού κόστους:**

Στο βασικό ερώτημα της επιλογής δεδομένων που έχουν μία συγκεκριμένη τιμή, η φάση της τοπικής επεξεργασίας είναι αυτή που επωμίζεται το μεγαλύτερο κόστος της επεξεργασίας. Οι διακομιστές θα πρέπει να απορρίψουν τα δεδομένα που δεν πληρούν τη συνθήκη της επιλογής ( $Salary=1400$ ) και να αποστείλουν τα υπόλοιπα. Τα δεδομένα που μεταφέρονται είναι μόνο αυτά που πληρούν τη συνθήκη, συνεπώς ο όγκος τους είναι μειωμένος. Αντίστοιχα μικρό είναι και το κόστος συγχώνευσης των δεδομένων στο συντονιστή. Συνεπώς, σε αλγορίθμους που σχετίζονται με τις επιλογές δεδομένων με συγκεκριμένες τιμές το κύριο βάρος επεξεργασίας βρίσκεται σε τοπικό επίπεδο. Η επιλογή να σταλούν όλα τα δεδομένα στο συντονιστή, με σκοπό να εκτελέσει ο ίδιος τη συγχώνευση όλων των δεδομένων του καταναμημένου συστήματος, για να βρει τα δεδομένα που πληρούν τη συνθήκη επιλογής, είναι φανερό ότι δεν έχει κανένα πλεονέκτημα σε σχέση με την παρούσα επιλογή, καθώς θα πρέπει να διακινηθεί στο δίκτυο μεγάλος όγκος δεδομένων. Θα πρέπει δηλαδή ο συντονιστής να συγκεντρώσει όλα τα δεδομένα, να τα επεξεργαστεί για να στείλει στη συνέχεια το σωστό τελικό αποτέλεσμα στο χρήστη. Τα δεδομένα που πρέπει, συνεπώς, να μεταφερθούν είναι πάρα πολλά συγκριτικά με το μέγεθος της καταναμημένης βάσης.

## 7.4 Ερώτημα 2<sup>ο</sup> - Υπολογισμός πεδίων τιμών

Πιο σύνθετο είναι το ερώτημα στο οποίο ένας χρήστης επιθυμεί έναν υπολογισμό τιμών σε δεδομένα που είναι αποθηκευμένα στο καταναμημένο σύστημα.

Έστω ότι ο χρήστης ζητάει να βρει τις συνολικές πωλήσεις που πραγματοποιήθηκαν σε ένα υποκατάστημα (π.χ. *Department 3*) από όλους τους πωλητές της εταιρείας.

*SUM Sales WHERE Department=3*

### Συνθήκη Α)





- Ο συντονιστής λαμβάνει το ερώτημα του χρήστη και στη συνέχεια το αποστέλλει σε όλους τους διακομιστές, έτσι ακριβώς όπως το έλαβε. Δηλαδή ζητάει το αποτέλεσμα της άθροισης κάθε διακομιστή που πληροί τους όρους του ερωτήματος.

#### I. Τοπική επεξεργασία

Κάθε διακομιστής θα πρέπει να συγκεντρώσει και να αθροίσει τα δεδομένα που πληρούν τους όρους του ερωτήματος. Συνεπώς, εκτελεί: *SUM Sales WHERE Department=3*.

<u>Employee ID</u>	Salary	<u>Department</u>	Sales (/month)
3	1100	3	8500
5	1100	3	6000
6	1400	3	17500
9	1100	3	7000
12	1300	3	13000
13	1400	3	13500
17	1300	3	15000

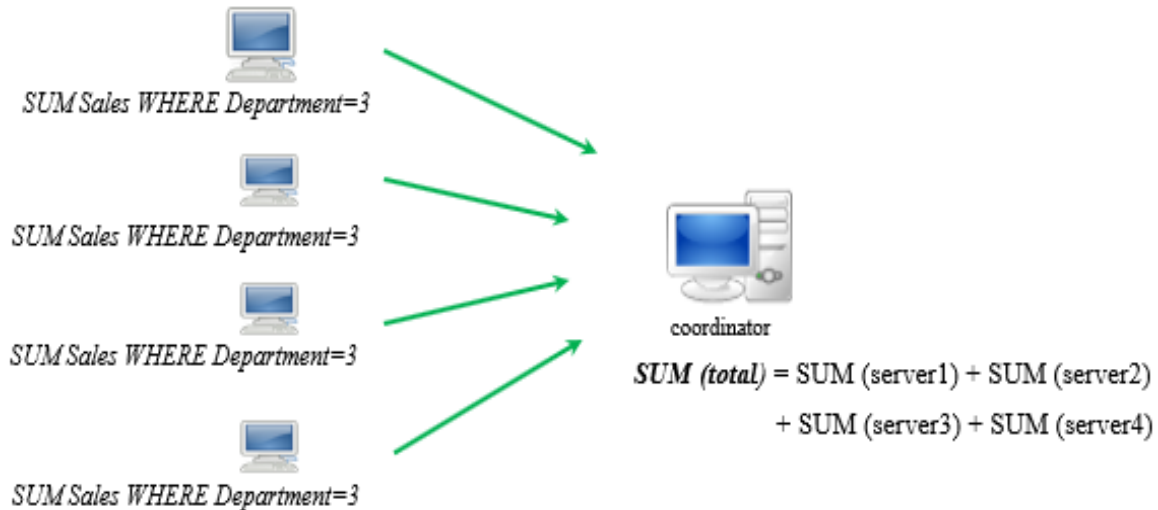
  

3	1100	3	8500	}  SUM
5	1100	3	6000	
6	1400	3	17500	}  SUM
9	1100	3	7000	
12	1300	3	13000	}  SUM
13	1400	3	13500	
17	1300	3	15000	}  SUM

Σχήμα 4 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 2<sup>ο</sup>, Συνθήκη Α)

## II. Μεταφορά δεδομένων

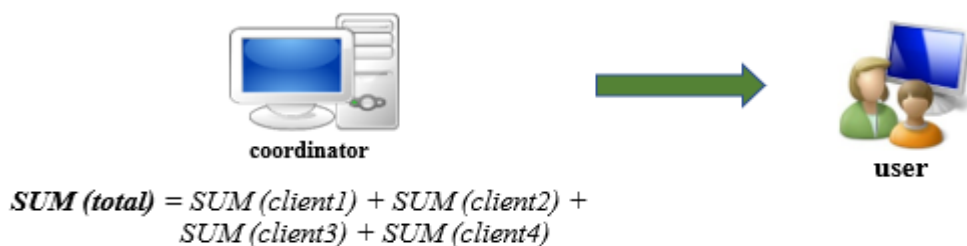
Κάθε διακομιστής επιστρέφει στο συντονιστή το αποτέλεσμα της άθροισης. Θα πρέπει όλοι οι διακομιστές να απαντήσουν στο ερώτημα για να μπορέσει ο συντονιστής να επεξεργαστεί τα αποτελέσματα. Αν δεν απαντήσουν όλοι, είναι αδύνατον να υπάρξει ορθό αποτέλεσμα.



Σχήμα 5 - Φάση μεταφοράς δεδομένων (Ερώτημα 2°, Συνθήκη Α)

## III. Συγχώνευση

Μόλις ο συντονιστής λάβει τις απαντήσεις από όλους τους διακομιστές, θα πρέπει να αθροίσει όλα τα αποτελέσματα που έλαβε για να προκύψει η τελική απάντηση του ερωτήματος που θα στείλει στον χρήστη.



Σχήμα 6 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 2°, Συνθήκη Α)

### Εκτίμηση συνολικού κόστους:

Από τις τρεις φάσεις καταναμημένης επεξεργασίας των δεδομένων το μεγαλύτερο κόστος εμπεριέχει η φάση της τοπικής επεξεργασίας, καθώς κάθε διακομιστής θα πρέπει να υπολογίσει το άθροισμα των δεδομένων που πληρούν τα κριτήρια του ερωτήματος. Το μικρότερο κόστος φαίνεται να έχει η διαδικασία μεταφοράς των δεδομένων, καθώς μόνο το αποτέλεσμα, ένας δηλαδή πολύ μικρός όγκος δεδομένων σε σχέση με τα δεδομένα που περιέχει η βάση, μεταφέρεται από τους διακομιστές στο συντονιστή. Η εκτίμηση αυτή παραμένει ίδια ανεξαρτήτως του όγκου δεδομένων

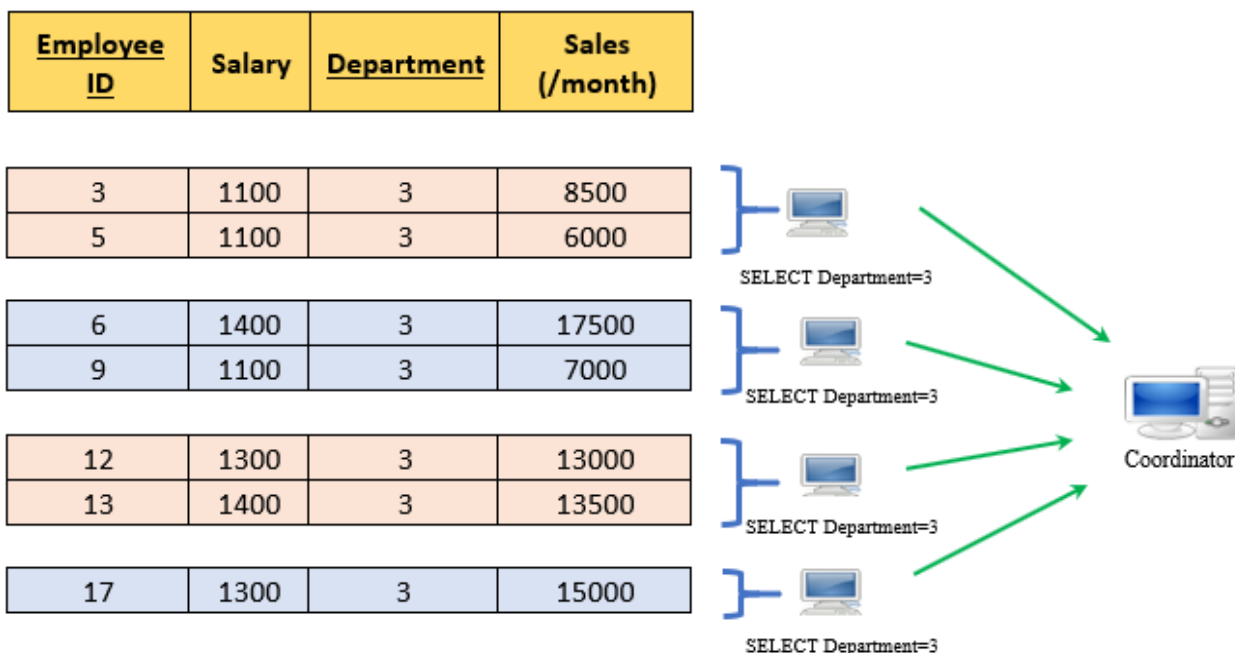
που θα έχουν να επεξεργαστούν οι διακομιστές. Όσο και να αυξηθεί το μέγεθος των δεδομένων που καταχωρούν οι διακομιστές το κόστος μεταφοράς τους θα παραμένει πολύ μικρό, καθώς μόνο το τελικό αποτέλεσμα προωθείται στο συντονιστή.

## Συνθήκη B)

- Ο συντονιστής λαμβάνει το ερώτημα του χρήστη αλλά επιλέγει να το μετασχηματίσει με σκοπό την τροποποίηση του κόστους στις φάσεις επεξεργασίας. Δεν λαμβάνει τα αποτελέσματα του ερωτήματος από κάθε διακομιστή, όπως στη *Συνθήκη A* και απλά τα συνενώνει αλλά δέχεται όλα τα δεδομένα τους, όσα πληρούν τα κριτήρια που έχουν τεθεί στο ερώτημα από τον χρήστη, και τα αθροίζει συνολικά. Στέλνει λοιπόν την εντολή για να λάβει τα δεδομένα από όλους τους διακομιστές που πληρούν τη συνθήκη  $Department=3$ .

### I. Τοπική επεξεργασία

Κάθε διακομιστής εκτελεί το ερώτημα  $SELECT Department=3$ . Απορρίπτονται όσες τιμές στο  $Department \neq 3$ . Δεν εκτελεί  $SUM Sales WHERE Department=3$ , όπως στην πρώτη συνθήκη, αλλά στέλνει στο συντονιστή όλες τις πλειάδες που πληρούν τη συνθήκη  $Department=3$  χωρίς καμία επιπλέον επεξεργασία.



Σχήμα 7 - Φάση τοπικής επεξεργασίας των δεδομένων (Ερώτημα 2°, Συνθήκη B)

## II. Μεταφορά δεδομένων

Ο συντονιστής λαμβάνει τα δεδομένα που πληρούν τη συνθήκη του ερωτήματος  $Department=3$ . Τα δεδομένα θα πρέπει να αποσταλούν από όλους τους διακομιστές για να ξεκινήσει την επεξεργασία τους, δηλαδή την άθροιση τους για να εξάγει το τελικό αποτέλεσμα.

## III. Συγχώνευση

Στη φάση της συγχώνευσης το κόστος επεξεργασίας των δεδομένων, δηλαδή την άθροιση των αποτελεσμάτων που έλαβε από τους διακομιστές, αναλαμβάνει να ολοκληρώσει ο συντονιστής, ο οποίος στη συνέχεια θα αποστείλει το οριστικό αποτέλεσμα του αθροίσματος στον χρήστη.



Σχήμα 8 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 2<sup>ο</sup>, Συνθήκη B)

### Εκτίμηση συνολικού κόστους:

Σε αντίθεση με τη χρήση του αλγορίθμου της *Συνθήκης A*, το μεγαλύτερο κόστος από τις τρεις φάσεις επεξεργασίας φέρει η μεταφορά των δεδομένων. Τα δεδομένα κάθε διακομιστή που πληρούν τους όρους του ερωτήματος αποστέλλονται στο συντονιστή, ο οποίος αναλαμβάνει την τελική τους επεξεργασία (το συνολικό άθροισμα), για την παραγωγή του οριστικού αποτελέσματος. Ο όγκος των δεδομένων που μετακινούνται είναι σημαντικά πιο μεγάλος σε σχέση με τη *Συνθήκη A* όπου μεταφέρεται μόνο το τελικό αποτέλεσμα κάθε διακομιστή. Επιπροσθέτως, ο συντονιστής έχει να επεξεργαστεί μεγάλο πλήθος δεδομένων καθυστερώντας την αποστολή του τελικού αποτελέσματος στο χρήστη. Στη φάση της τοπικής επεξεργασίας οι διακομιστές δεν επιβαρύνονται με επιπρόσθετους υπολογισμούς των δεδομένων, απλά αναζητούν στη βάση τους τα σωστά δεδομένα, όσα δηλαδή πληρούν τους όρους του ερωτήματος, και τα αποστέλλουν. Στην περίπτωση όμως μελλοντικής διεύρυνσης του DDBS και του δικτύου, θα αυξάνεται αναλογικά τόσο το κόστος της επικοινωνίας μεταξύ των διακομιστών και του συντονιστή όσο και το κόστος συγχώνευσης/επεξεργασίας των δεδομένων στο συντονιστή. Το κόστος της τοπικής εξεργασίας θα αυξάνεται, επίσης, αλλά όχι σε σημαντικό βαθμό καθώς οι διακομιστές θα εκτελούν μόνο την εντολή αναζήτησης των απαιτούμενων δεδομένων και όχι κάποια επιπλέον επεξεργασία.

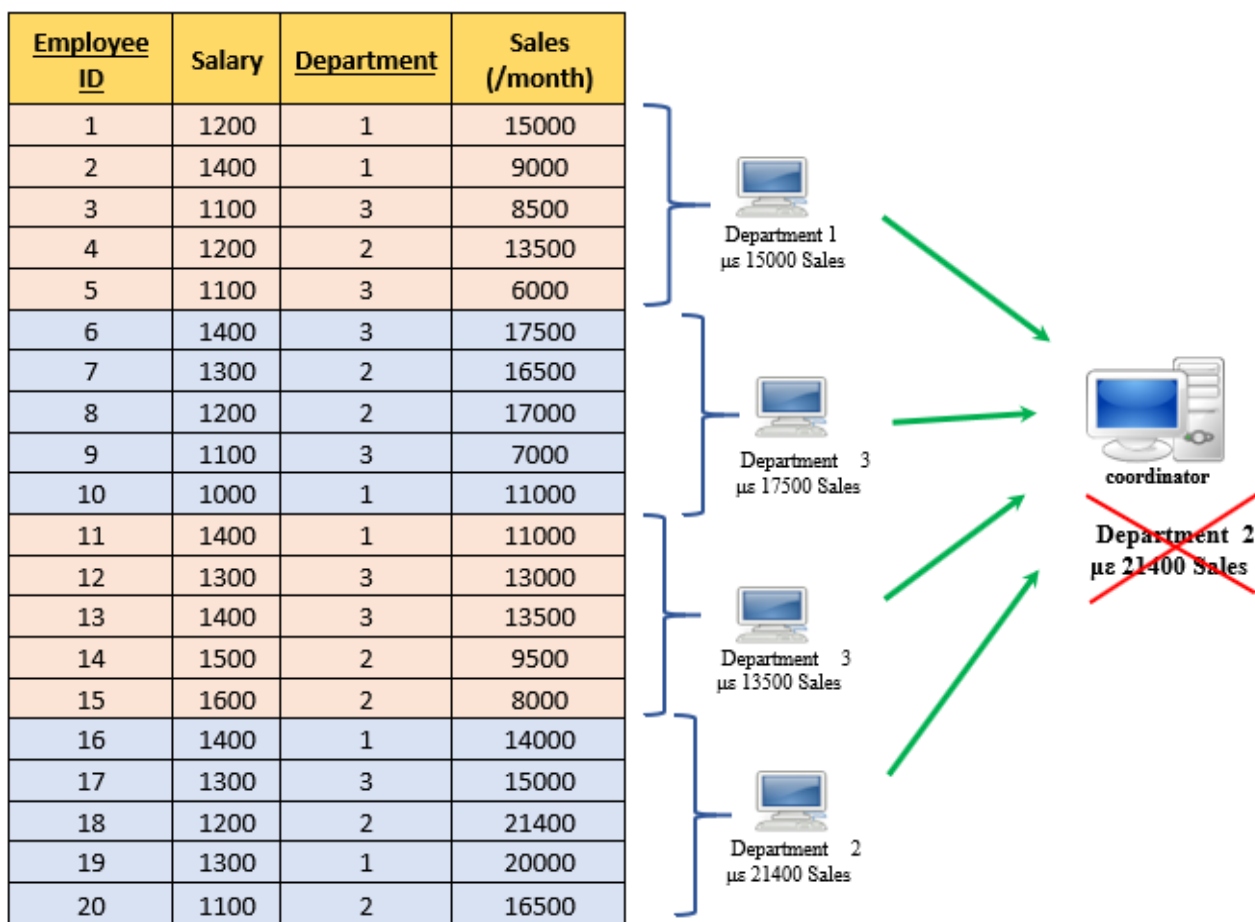


## 7.5 Ερώτημα 3<sup>ο</sup> - Αναζήτηση καλύτερης τιμής (top K)

Καθώς θα προχωράμε στην παρουσίαση ερωτημάτων τόσο το κόστος σε κάθε φάση επεξεργασίας, όσο και το συνολικό κόστος θα αυξάνονται ως αποτέλεσμα των πιο σύνθετων διεργασιών που θα πρέπει να πραγματοποιηθούν. Ένα αρκετά σύνηθες ερώτημα σε ένα παράλληλο και κατανεμημένο σύστημα αποτελεί η αναζήτηση της καλύτερης τιμής μίας ομάδας δεδομένων (top K).

Έστω ότι ένας χρήστης επιθυμεί να βρει το υποκατάστημα που έχει πετύχει το μεγαλύτερο ποσό σε πωλήσεις ανά μήνα. Στην περίπτωση αυτή θα δούμε ότι χρειάζεται ιδιαίτερη προσοχή στον τρόπο εκτέλεσης ενός αλγορίθμου καθώς αυτός μπορεί να οδηγηθεί σε λάθος αποτελέσματα.

Αν π.χ. ο συντονιστής απλά ζητήσει από κάθε διακομιστή το υποκατάστημα με τις περισσότερες πωλήσεις *SELECT MAX (Sales)*, όπως θα δούμε παρακάτω στο σχήμα, ο συντονιστής θα δώσει λάθος τελική απάντηση:



Σχήμα 9 - Εκτέλεση αλγορίθμου που οδηγεί σε λανθασμένα αποτελέσματα (Ερώτημα 3<sup>ο</sup>)

Οι απαντήσεις που θα λάβει ο συντονιστής από τους διακομιστές και κατ' επέκταση ο τελικός χρήστης θα είναι **λάθος**. Συνεπώς, θα πρέπει να δοθεί ιδιαίτερη προσοχή όταν οι αλγόριθμοι θα



πρέπει να διαχειριστούν πιο σύνθετα ερωτήματα έτσι ώστε να είναι βέβαιο ότι ο χρήστης θα λάβει σωστά αποτελέσματα.

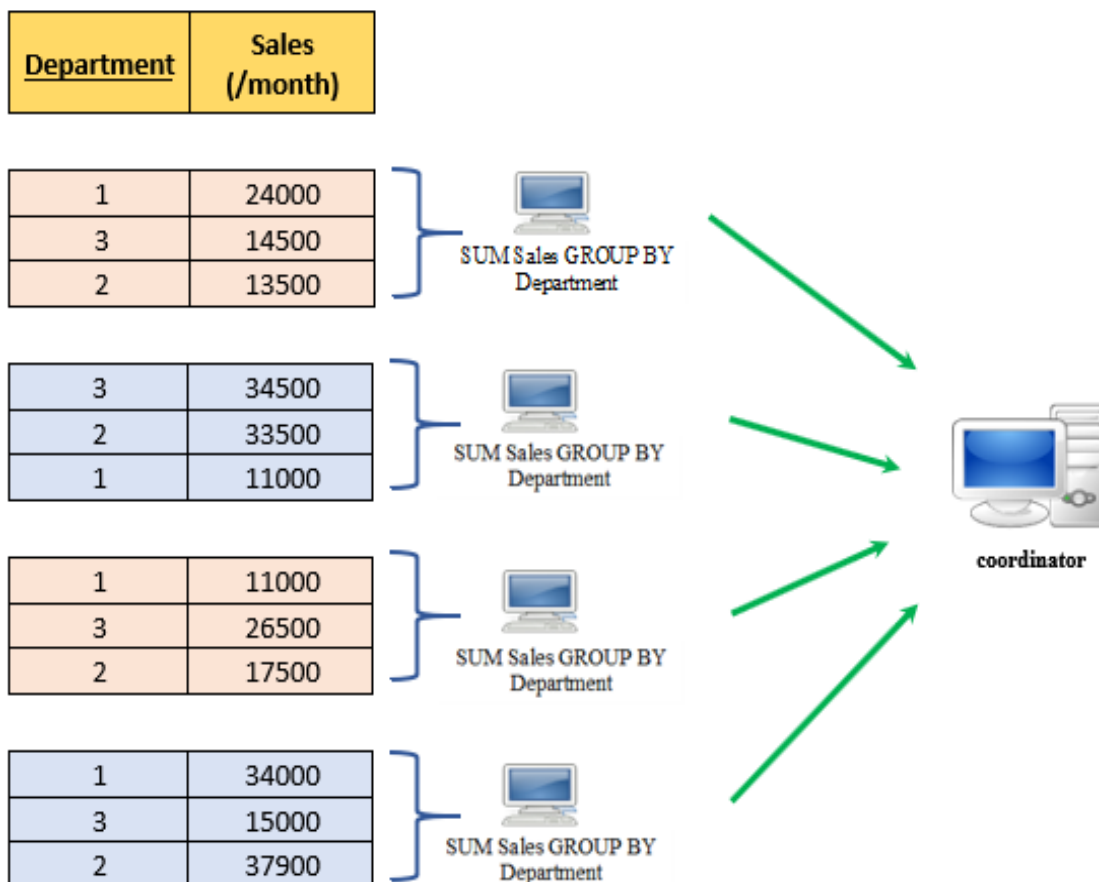
Στο ερώτημα λοιπόν του χρήστη ποιο υποκατάστημα έχει πραγματοποιήσει τις περισσότερες πωλήσεις (/μήνα), θα πρέπει ο αλγόριθμος να ακολουθήσει συγκεκριμένα βήματα ώστε να είμαστε βέβαιοι για την ορθή απάντηση. Μπορούμε να ακολουθήσουμε μία από τις δύο παρακάτω επιλογές οι οποίες σε κάθε φάση επεξεργασίας παρουσιάζουν διαφορετικά κόστη.

## Συνθήκη Α)

- Ο συντονιστής λαμβάνει το ερώτημα του χρήστη *SELECT MAX (Sales)* και το μετασχηματίζει σε *SUM Sales GROUP BY Department* με σκοπό να λάβει τα σωστά αποτελέσματα σχετικά την ανεύρεση του υποκαταστήματος που έχει πραγματοποιήσει τις περισσότερες πωλήσεις.

### I. Τοπική επεξεργασία

Κάθε διακομιστής εκτελεί την εντολή *SUM Sales GROUP BY Department* για να υπολογίσει τις συνολικές πωλήσεις ανά υποκατάστημα και αποστέλλει την απάντηση στον συντονιστή.



Σχήμα 10 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 3°, Συνθήκη Α)

## II. Μεταφορά δεδομένων

Το σύνολο των δεδομένων που μεταφέρονται από τους διακομιστές στο συντονιστή έχουν περιοριστεί αρκετά, λόγω της επεξεργασίας που έχουν υποστεί σε τοπικό επίπεδο, στην προηγούμενη φάση. Κάθε διακομιστής αποστέλλει μόνο το αποτέλεσμα των συνολικών πωλήσεων ανά υποκατάστημα.

## III. Συγχώνευση

Ο συντονιστής λαμβάνει από κάθε διακομιστή τα αποτελέσματα της άθροισης των πωλήσεων ανά υποκατάστημα. Στη συνέχεια θα πρέπει να αθροίσει και αυτός ανά υποκατάστημα τα αποτελέσματα που έλαβε, να τα ταξινομήσει, για να βρει ποιο είχε τις υψηλότερες πωλήσεις, και να στείλει την τελική απάντηση στον χρήστη.



Σχήμα 11 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 3<sup>ο</sup>, Συνθήκη Α)

### Εκτίμηση συνολικού κόστους:

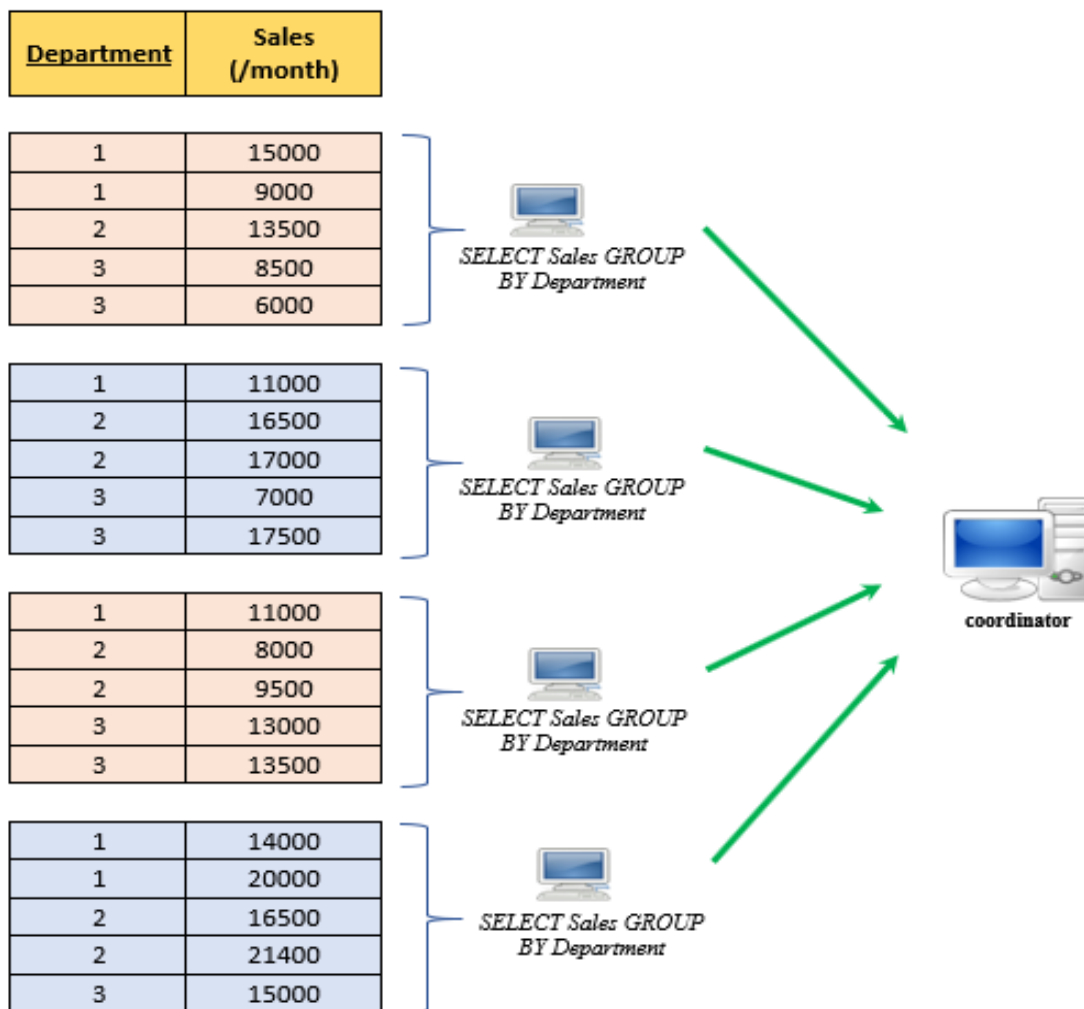
Στη συγκεκριμένη *Συνθήκη* υπάρχει μια ισορροπημένη κατανομή των εργασιών, δηλαδή των δεδομένων που θα πρέπει να διαχειριστούν τόσο οι διακομιστές όσο και ο συντονιστής. Τα δεδομένα που πρέπει να μεταδοθούν για να διαχειριστεί στη συνέχεια ο συντονιστής έχουν περιοριστεί αρκετά λόγω του φιλτραρίσματος (ταξινόμηση και άθροισμα πωλήσεων ανά υποκατάστημα) που έχουν υποστεί σε τοπικό επίπεδο (από τους διακομιστές). Επιπλέον, και το μέγεθος των δεδομένων που μεταφέρονται είναι περιορισμένο καθώς μόνο το αποτελέσματα της άθροισης διακινείται, περιορίζοντας σε μεγάλο βαθμό το κόστος της συγκεκριμένης φάσης.

## Συνθήκη Β)

- Το αρχικό ερώτημα παραμένει το ίδιο όπως και στη *Συνθήκη Α*. Ο συντονιστής λαμβάνει το ερώτημα του χρήστη *SELECT MAX Sales* το οποίο όμως το μετασχηματίζει σε *SELECT Sales GROUP BY Department* με σκοπό να λάβει από όλους τους διακομιστές τα στοιχεία των πωλήσεων ταξινομημένα ανά υποκατάστημα.

### I. Τοπική επεξεργασία

Κάθε διακομιστής εκτελεί την εντολή *SELECT Sales GROUP BY Department* για να επιλέξει και να ταξινομήσει τις πωλήσεις κάθε υποκαταστήματος που είναι αποθηκευμένα στη βάση του. Στη συνέχεια στέλνει τα αποτελέσματα στο συντονιστή.



Σχήμα 12 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 3°, Συνθήκη Β)

## II. Μεταφορά δεδομένων

Ο συντονιστής λαμβάνει το σύνολο των δεδομένων που πληρούν τους όρους του ερωτήματος. Η μόνη επεξεργασία που έχουν υποστεί τα δεδομένα στους διακομιστές είναι η ταξινόμησή τους ανά υποκατάστημα.

## III. Συγχώνευση

Ο συντονιστής έχει λάβει τα δεδομένα και πρέπει να εκτελέσει μία σειρά εντολών για να αποστείλει την τελική απάντηση στο ερώτημα του χρήστη. Θα πρέπει να προσθέσει τις πωλήσεις που έλαβε από όλους τους διακομιστές ανά υποκατάστημα και στη συνέχεια αφού τα ταξινομήσει, βάσει των πωλήσεών τους, να στείλει ως απάντηση στο χρήστη το υποκατάστημα με τις υψηλότερες πωλήσεις του μήνα.



Σχήμα 13 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 3°, Συνθήκη B)

### Εκτίμηση συνολικού κόστους:

Στη *Συνθήκη B*, στο ερώτημα αναζήτησης της καλύτερης τιμής παρατηρούμε ότι ενώ είναι αποτελεσματικός ο αλγόριθμος, αφού δίνει το σωστό αποτέλεσμα, είναι πολύ λιγότερο αποδοτικός. Το κόστος της τοπικής επεξεργασίας στους διακομιστές είναι χαμηλό, καθώς το μόνο που έχουν να κάνουν είναι μία ταξινόμηση των δεδομένων τους και να τα αποστείλουν στο συντονιστή. Είναι όμως εξαιρετικά μεγάλο το κόστος επικοινωνίας, καθώς στο δίκτυο θα πρέπει να μεταφερθεί μεγάλος όγκος δεδομένων, συγκεκριμένα το σύνολο των δεδομένων κάθε διακομιστή, ενώ και στη τελική φάση της συγχώνευσης ο συντονιστής έχει να επεξεργαστεί το σύνολο των δεδομένων της καταναμημένης βάσης.

## 7.6 Ερώτημα 4<sup>ο</sup> - Ταξινόμηση αποτελεσμάτων

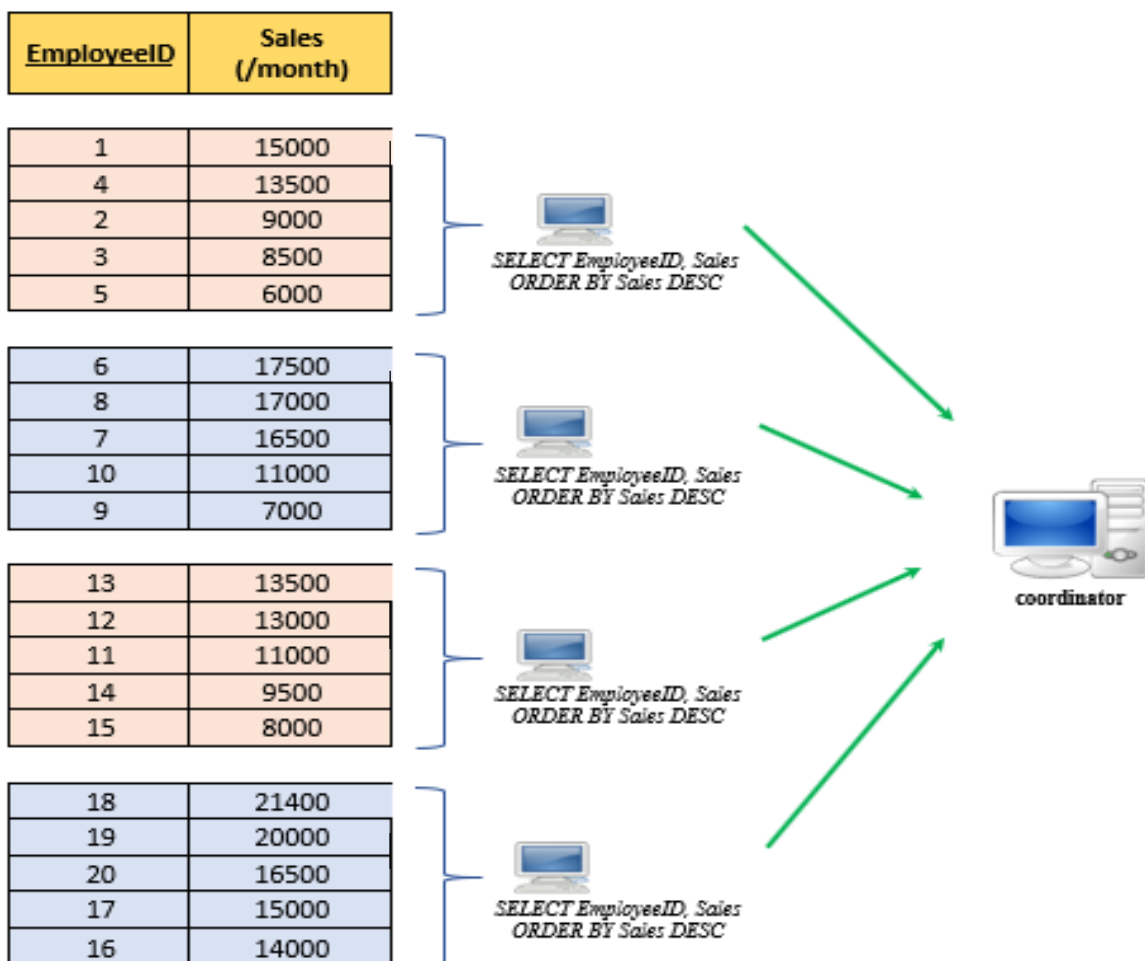
Ένα ακόμα πολύ σύνθηες ερώτημα που θέτεται από χρήστες σε DDBS είναι η ταξινόμηση των δεδομένων βάσει κάποιας τιμής τους. Χρησιμοποιώντας το δικό μας παράδειγμα, έστω ότι ένας χρήστης ζητάει για στατιστικούς λόγους την ταξινόμηση των πωλητών της εταιρείας (σε φθίνουσα σειρά), σύμφωνα με τις πωλήσεις (/μήνα) που έχουν πραγματοποιήσει. Ο χρήστης λοιπόν στέλνει το ερώτημα *SELECT EmployeeID, Sales ORDER BY Sales DESC*.

### Συνθήκη Α)

- Ο συντονιστής λαμβάνει το ερώτημα του χρήστη και το στέλνει με τη σειρά του, χωρίς καμία τροποποίηση, στους διακομιστές.

#### 1. Τοπική επεξεργασία

Κάθε διακομιστής εκτελεί την εντολή *SELECT EmployeeID, Sales ORDER BY Sales DESC* για την επιλογή και ταξινόμηση των πωλητών, σύμφωνα με τις πωλήσεις που πραγματοποίησαν και αποστέλλει τα αποτελέσματα στον συντονιστή.



Σχήμα 14 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 4<sup>ο</sup>, Συνθήκη Α)

## II. Μεταφορά δεδομένων

Οι διακομιστές έχουν επεξεργαστεί το σύνολο των δεδομένων τους που σχετίζονται με τα πεδία τιμών *EmproleeID* και *Sales* και αποστέλλουν τα αποτελέσματα ταξινομημένα σύμφωνα με τους όρους του ερωτήματος.

## III. Συγχώνευση

Ο συντονιστής δέχεται διαδοχικά τις απαντήσεις των διακομιστών. Καθώς λαμβάνει ταξινομημένα τα δεδομένα από κάθε έναν, μπορεί να τα αποστείλει άμεσα στον χρήστη, χωρίς να περιμένει να ολοκληρωθεί η διαδικασία από όλους τους διακομιστές. Συνεπώς, ο χρήστης λαμβάνει άμεσα όσα αποτελέσματα έχει στη διάθεσή του ο συντονιστής.



Σχήμα 15 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 4°, Συνθήκη Α)

### Εκτίμηση συνολικού κόστους:

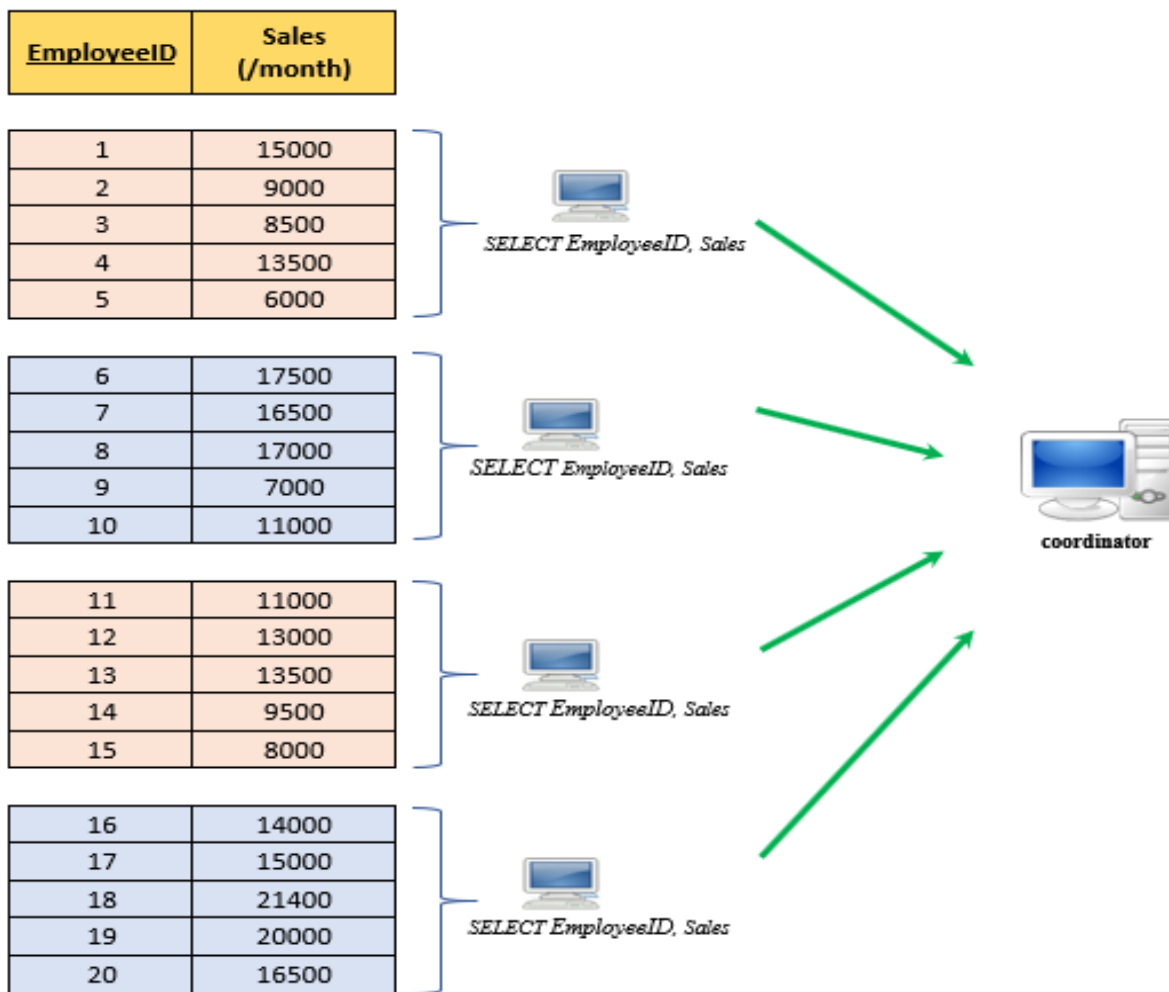
Επιλέγοντας τον συγκεκριμένο τρόπο απάντησης του ερωτήματος, το μεγαλύτερο κόστος από τις 3 φάσεις επεξεργασίας εμπεριέχει η φάση της μεταφοράς των δεδομένων. Στην πρώτη φάση της τοπικής επεξεργασίας κάθε διακομιστής θα πρέπει να ταξινομήσει το σύνολο των δεδομένων του για να τα αποστείλει στον συντονιστή. Καθώς η επεξεργασία αποτελείται μόνο από την ταξινόμηση των δεδομένων, το κόστος είναι ιδιαίτερα χαμηλό. Στη δεύτερη φάση, όμως, ο όγκος της μεταφοράς των δεδομένων είναι αρκετά μεγάλος καθώς θα πρέπει να μεταδοθούν το σύνολο των σχετικών δεδομένων χωρίς να έχει “κλαδευτεί” μέρος αυτού που σημαίνει ότι το δίκτυο θα επιβαρυνθεί σε μεγάλο βαθμό. Στο τελικό στάδιο της συγχώνευσης ο συντονιστής μπορεί να αποστείλει άμεσα τα αποτελέσματα κάθε διακομιστή. Δεν χρειάζεται να κάνει κανενός είδους επιπλέον υπολογισμούς ή επεξεργασία στα δεδομένα που έλαβε. Συνεπώς αποστέλλει στον χρήστη άμεσα τις απαντήσεις από κάθε διακομιστή. Τα τελικά αποτελέσματα ταξινομούνται από τον χρήστη σε πραγματικό χρόνο έτσι όπως λαμβάνονται.

## Συνθήκη Β)

- Ο συντονιστής λαμβάνει το ερώτημα του χρήστη και το αποστέλλει με τη σειρά του στους διακομιστές.

### I. Τοπική επεξεργασία

Κάθε διακομιστής εκτελεί την εντολή *SELECT EmployeeID, Sales* επιλέγοντας τα δεδομένα που θα στείλει προς επεξεργασία στον συντονιστή.



Σχήμα 16 - Φάση τοπικής επεξεργασίας δεδομένων (Ερώτημα 4°, Συνθήκη Β)

### II. Μεταφορά δεδομένων

Όλα τα δεδομένα που σχετίζονται με τον κωδικό πωλητή και τις πωλήσεις του (/μήνα) μεταφέρονται στο συντονιστή όπως ακριβώς είναι αποθηκευμένα στην κατακευμαμένη βάση, χωρίς επεξεργασία.

### III. Συγχώνευση

Ο συντονιστής αναμένει από όλους τους διακομιστές την αποστολή όλων των σχετιζόμενων δεδομένων για να προχωρήσει στην ταξινόμησή τους για την παραγωγή του τελικού αποτελέσματος.



Σχήμα 17 - Φάση συγχώνευσης των δεδομένων (Ερώτημα 4<sup>ο</sup>, Συνθήκη B)

#### **Εκτίμηση συνολικού κόστους:**

Σε αντίθεση με τον αλγόριθμο της *Συνθήκης Α* η φάση επεξεργασίας με το μεγαλύτερο κόστος είναι αυτή της μεταφοράς των δεδομένων. Στην πρώτη φάση της τοπικής επεξεργασίας οι διακομιστές μεταφέρουν αυτούσια, χωρίς επεξεργασία, τα δεδομένα που έχει ζητήσει ο συντονιστής. Στο στάδιο της μεταφοράς των δεδομένων όλα τα σχετιζόμενα δεδομένα των διακομιστών μεταδίδονται χωρίς να υποστούν καμία επεξεργασία που θα μείωνε το φόρτο μεταφοράς τους ή επεξεργασίας τους στην επόμενη φάση. Στην τελική φάση της συγχώνευσης ο συντονιστής θα πρέπει να αναμένει τον πιο αργό διακομιστή έτσι ώστε να μπορέσει να ταξινομήσει το σύνολο των απεσταλμένων δεδομένων και να προχωρήσει στην παραγωγή του τελικού αποτελέσματος.

## 7.7 Συμπεράσματα

Εξετάζοντας με προσοχή τα παραπάνω παραδείγματα, έγιναν οι εξής διαπιστώσεις αναφορικά με τα κόστη των φάσεων επεξεργασίας των δεδομένων. Σε αλγορίθμους απλής επιλογής δεδομένων (ερώτημα 1<sup>ο</sup>) το κύριο βάρος του κόστους επεξεργασίας βρίσκεται σε τοπικό επίπεδο, στους διακομιστές. Πιο σύνθετα ερωτήματα που απαιτούν επιπλέον υπολογισμούς των δεδομένων της κατανεμημένης βάσης χρειάζονται προσοχή κατά τον σχεδιασμό της εκτέλεσης των αλγορίθμων έτσι ώστε να μην οδηγείται το σύστημα σε λανθασμένο αποτέλεσμα. Επιπλέον, όσο περισσότερο ένα ερώτημα επεξεργάζεται σε τοπικό επίπεδο, τόσο λιγότερο είναι το κόστος που επωμίζονται οι φάσεις της επικοινωνίας και της συγχώνευσης των δεδομένων. Ο καταμερισμός των εργασιών σε κάθε μία από τις τρεις φάσεις επεξεργασίας θα πρέπει να είναι όσο το δυνατόν ισομερής, σύμφωνα πάντα με τα χαρακτηριστικά του κατανεμημένου συστήματος. Στην τελική φάση της συγχώνευσης ο συντονιστής θα πρέπει να δέχεται όσο το δυνατόν μικρότερο όγκο δεδομένων και μην



επιφορτίζεται με ακατέργαστα δεδομένα που χρειάζονται χρονοβόρους υπολογισμούς και πόρους, αυξάνοντας το συνολικό κόστος διεκπεραίωσης του ερωτήματος. Όσο λιγότερη επεξεργασία έχουν υποστεί τα δεδομένα σε τοπικό επίπεδο (στη πρώτη φάση), τόσο περισσότερο αυξάνεται το κόστος στην τελική φάση της συγχώνευσης των δεδομένων. Είναι, συνεπώς, σημαντικά αποδοτικότερο οι διεργασίες να μοιράζονται σε τοπικό επίπεδο στους υπολογιστές του δικτύου και τα δεδομένα να μεταφέρονται φιλτραρισμένα για συγχώνευση, από το να αναλάβει ο συντονιστής την επεξεργασία μεγάλου όγκου δεδομένων για την παραγωγή του τελικού αποτελέσματος. Τέλος, θα πρέπει τα δεδομένα που μεταφέρονται κατά τη φάση της επικοινωνίας να είναι όσο το δυνατόν περιορισμένα, συγκρινόμενα πάντα με το μέγεθος της κατανεμημένης βάσης.

Στον παρακάτω πίνακα παρουσιάζονται συγκεντρωτικά οι φάσεις επεξεργασίας των ερωτημάτων που εξετάσαμε, εντοπίζοντας τις φάσεις εκείνες κατά τις οποίες παρουσιάζεται αύξηση ή μείωση του κόστους επεξεργασίας του ερωτήματος.

			<b>Φάσεις επεξεργασίας</b>		
	<b>Ερωτήματα</b>	Συνθήκη	<b>Τοπική επεξεργασία</b>	<b>Μεταφορά δεδομένων</b>	<b>Συγχώνευση</b>
1	Επιλογής τιμών				Άμεση αποστολή αποτελεσμάτων
2	Υπολογισμός πεδίων τιμών	A			Συνένωση + επεξεργασία
		B			Συνένωση + επεξεργασία
3	Αναζήτηση καλύτερης τιμής (top k)	A			Συνένωση + επεξεργασία
		B			Συνένωση + επεξεργασία
4	Ταξινόμηση αποτελεσμάτων	A			Άμεση αποστολή αποτελεσμάτων
		B			Συνένωση + επεξεργασία

Αυξημένο κόστος

Μειωμένο κόστος

Πίνακας 1 – Συγκεντρωτικός πίνακας παρουσίασης κόστους φάσεων δεδομένων βάσει των ερωτημάτων που χρησιμοποιήθηκαν

# 8

## Σύγχρονες εφαρμογές παράλληλης και κατανεμημένης διαχείρισης δεδομένων

Όπως αναφέρθηκε και παραπάνω (κεφάλαιο 2.4), εφαρμογές της παράλληλης και κατανεμημένης διαχείρισης δεδομένων μπορούμε να βρούμε σε πολλές σύγχρονες δραστηριότητες που σχετίζονται με υπολογιστικά και τηλεπικοινωνιακά συστήματα. Κάθε μία εφαρμογή ορίζεται από διαφορετικά χαρακτηριστικά, τρόπο λειτουργίας και απαιτήσεις. Συνεπώς, για κάθε μία από αυτές θα πρέπει οι αλγόριθμοι να σχεδιαστούν σύμφωνα με τις αξιώσεις του συστήματος ώστε να επιτευχθεί η αποτελεσματικότητα και αποδοτικότητα που απαιτείται.

Το παράδειγμα που χρησιμοποιήθηκε στο 7<sup>ο</sup> κεφάλαιο παρουσίαζε τον τρόπο με τον οποίο υλοποιούνται βασικά ερωτήματα σε ένα γενικό, αφηρημένο κατανεμημένο σύστημα αλλά και τη διαδικασία διαμόρφωσης του κόστους σε κάθε φάση επεξεργασίας, ανάλογα με τον αλγόριθμό και τη συνθήκη του τρόπου ολοκλήρωσης του ερωτήματος. Το παράδειγμα αυτό, ίσως, δεν είναι αρκετό για να αποδειχθεί με ακρίβεια ποια φάση επεξεργασίας σε κάθε ερώτημα είναι σημαντικότερο να έχει το χαμηλότερο κόστος για να επιτευχθεί η μέγιστη αποδοτικότητα. Αν, όμως, γνωρίζουμε το κατανεμημένο σύστημα που θα χρησιμοποιηθεί (αρχιτεκτονική, σχεδιασμός, χαρακτηριστικά, ιδιότητες), υπάρχει η δυνατότητα μία γενικής πρόγνωσης του κόστους στην οποία μπορούμε να προσδιορίσουμε ποια φάση επεξεργασίας αναμένεται να έχει υψηλό κόστος και ποια λιγότερο. Μπορούμε δηλαδή μέσω μίας γενικής κατηγοριοποίησης των εφαρμογών των παράλληλων και κατανεμημένων συστημάτων να εντοπίσουμε ποια από τις τρεις φάσεις επεξεργασίας των δεδομένων είναι πιθανότερο να έχει υψηλότερο κόστος και ποια χαμηλότερο.

### 8.1 Κατηγοριοποίηση των σύγχρονων εφαρμογών.

Μια γενική κατηγοριοποίηση των εφαρμογών που βασίζονται σε παράλληλα και κατανεμημένα συστήματα μπορεί να πραγματοποιηθεί βάσει της αρχιτεκτονικής σχεδιασμού και του τρόπου επικοινωνίας μεταξύ των αλληλοσυνδεδεμένων κόμβων του δικτύου ενός συστήματος, ανεξαρτήτως του όγκου δεδομένων που έχει να διαχειριστεί.

- **Δίκτυα Τηλεπικοινωνιών** (Δίκτυα τηλεφωνίας και κινητής τηλεφωνίας, δίκτυα υπολογιστών όπως το internet και ασύρματα δίκτυα αισθητήρων).

Βασικό χαρακτηριστικό της κατηγορίας αποτελεί το γεγονός ότι η σύνδεση μεταξύ των κόμβων που αποτελούν το κατανεμημένο σύστημα πραγματοποιείται μέσω του δικτύου επικοινωνιών. Πρωταρχικό στόχο των αλγορίθμων διαχείρισης των δεδομένων σε ένα κατανεμημένο σύστημα το οποίο βασίζει την επικοινωνία μεταξύ των διασυνδεδεμένων

κόμβων σε δίκτυα τηλεπικοινωνιών θα πρέπει να αποτελεί η ελαχιστοποίηση του κόστους της φάσης μεταφοράς των δεδομένων. Τα δίκτυα επικοινωνιών, πάρα τα τεχνολογικά άλματα που έχουν πραγματοποιηθεί για την αύξηση της ταχύτητας των δεδομένων, καθώς και για τη σταθερότητα του δικτύου κατά τη μεταφοράς τους, δεν μπορούν να εγγυηθούν πάντα σταθερή υψηλή απόδοση. Επιπροσθέτως, υπάρχει και περιορισμός στο όγκο των δεδομένων που μπορούν να μεταφερθούν χωρίς να υπερφορτωθεί το δίκτυο του συστήματος. Ενώ, παράλληλα, τίθεται και το θέμα του οικονομικού κόστους μεταφοράς των δεδομένων καθώς σε πολλές περιπτώσεις οι εταιρείες τηλεπικοινωνιών βάζουν όριο στον όγκο μεταφοράς τους, το οποίο εάν ξεπεραστεί υπάρχει πρόσθετη οικονομική επιβάρυνση. Συνεπώς, ενός αλγόριθμος θα πρέπει να είναι σχεδιασμένος έτσι ώστε το μέγιστο των διεργασιών να πραγματοποιείται στη φάση της τοπικής επεξεργασίας, μειώνοντας όσο είναι δυνατόν το κόστος της φάσης της μεταφοράς των δεδομένων για να είναι αποτελεσματικός σε συστήματα που βασίζονται στις τηλεπικοινωνίες.

- **Εφαρμογές διαδικτύου**

Η συγκεκριμένη κατηγορία αφορά σε κατανεμημένες βάσεις δεδομένων και κατανεμημένα συστήματα βάσεων δεδομένων, στον παγκόσμιο ιστό (*World Wide Web*), στα δίκτυα P2P (*Peer to Peer*), στα online βιντεοπαιχνίδια στα οποία συμμετέχουν πολλοί παίχτες ταυτόχρονα και στα κατανεμημένα συστήματα επεξεργασίας πληροφοριών, όπως είναι τα τραπεζικά συστήματα και οι εταιρείες κρατήσεων αεροπορικών εισιτηρίων. Το σύστημα διαχείρισης των κατανεμημένων δεδομένων θα πρέπει να είναι σχεδιασμένο έτσι ώστε οι φάσεις της τοπικής επεξεργασίας και της μεταφοράς των δεδομένων να είναι αυτές που θα είναι περισσότερο επιφορτισμένες με την ολοκλήρωση ενός ερωτήματος. Οι αλληλοσυνδεδεμένοι κόμβοι θα πρέπει να είναι επιφορτισμένοι με την επεξεργασία των δεδομένων στη φάση της τοπικής επεξεργασίας, όσον το δυνατόν σε μεγαλύτερο βαθμό, προκειμένου να ελαχιστοποιηθεί το κόστος της συγχώνευσης κατά την τελική φάση επεξεργασίας των δεδομένων. Παράλληλα, αυξημένο φορτίο μεταφοράς δεδομένων μπορεί να υποστηρίξει και η φάση της επικοινωνίας καθώς, όπως αναφέρθηκε, οι σύγχρονες τεχνολογίες (VDSL, Fiber) έχουν τη δυνατότητα μεταφοράς μεγάλου όγκου δεδομένων με ταχύτητα, σταθερότητα και αξιοπιστία.

- **Δωμάτιο διακομιστών (servers room).**

Στην περίπτωση που τα κατανεμημένα δεδομένα βρίσκονται σε ένα δωμάτιο διακομιστών (όπως π.χ. σε μία μεγάλη επιχείρηση ή οργανισμό) η διαχείριση του κόστους στις φάσεις επεξεργασίας των δεδομένων αλλάζει. Για να επιτευχθεί η μέγιστη απόδοση θα πρέπει το ερώτημα να επεξεργάζεται σε τοπικό επίπεδο, σε κάθε κόμβο, και τα αποτελέσματα να μεταφέρονται για συγχώνευση. Λόγω της πολύ μικρής απόστασης μεταξύ των διακομιστών και της ενσύρματης επικοινωνίας μεταξύ των συνδεδεμένων κόμβων η φάση της μεταφοράς των δεδομένων μπορεί να δεχτεί με ασφάλεια και ταχύτητα σημαντικά μεγαλύτερο όγκο δεδομένων σε σχέση με άλλες εφαρμογές των κατανεμημένων συστημάτων. Συνεπώς, το σύστημα θα πρέπει να είναι σχεδιασμένο έτσι ώστε το

μεγαλύτερο βάρος της επεξεργασίας του ερωτήματος να πραγματοποιείται στις δύο πρώτες φάσεις επεξεργασίας και λιγότερο στη φάση της συγχώνευσης.

• Σε ελέγχους διαδικασιών σε πραγματικό χρόνο.

Βιομηχανικά συστήματα ελέγχου και συστήματα διαχείρισης εναέριας κυκλοφορίας αεροσκαφών χρησιμοποιούν καταναμημένα συστήματα που απαιτούν των έλεγχο διαδικασιών σε πραγματικό χρόνο. Σε καταναμημένα συστήματα που πραγματοποιούνται έλεγχοι διαδικασιών σε πραγματικό χρόνο είναι πολύ σημαντικό ο χρήστης να λαμβάνει το αποτέλεσμα άμεσα, το συντομότερο δυνατόν. Θα πρέπει τα δεδομένα να είναι καταναμημένα ισομερώς σε όλους τους κόμβους του δικτύου και να επεξεργάζονται σε τοπικό επίπεδο. Η φάση της συγχώνευσης δεν θα πρέπει να επιβαρύνεται με διαδικασίες ή επεξεργασία δεδομένων που απαιτούν αυξημένο χρόνο για την αποστολή του τελικού αποτελέσματος. Θα πρέπει δηλαδή κατά τη φάση της συγχώνευσης να πραγματοποιείται όσο το δυνατόν λιγότερη επεξεργασία δεδομένων, όσο φυσικά το επιτρέπει η φύση του ερωτήματος, ελαχιστοποιώντας το χρόνο από τη στιγμή που θα σταλεί ένα ερώτημα έως τη στιγμή που θα απαντηθεί. Είναι σημαντικό να μεταφέρεται άμεσα στον χρήστη το αποτέλεσμα κάθε κόμβου. Απαιτείται η τελική φάση της συγχώνευσης να έχει ελάχιστη συνεισφορά στην επεξεργασία των δεδομένων.

Στον ακόλουθο πίνακα παρουσιάζονται συγκεντρωτικά οι φάσεις επεξεργασίας των σύγχρονων εφαρμογών των παράλληλων και καταναμημένων συστημάτων με το επιθυμητό κόστος ανά φάση επεξεργασίας.

		<b>Φάσεις επεξεργασίας</b>		
	<b>Εφαρμογές</b>	<b>Τοπική επεξεργασία</b>	<b>Μεταφορά δεδομένων</b>	<b>Συγχώνευση</b>
<b>1</b>	Δίκτυα τηλεπικοινωνιών			
<b>2</b>	Εφαρμογές διαδικτύου			
<b>3</b>	Server rooms			
<b>4</b>	Έλεγχοι διαδικασιών σε πραγματικό χρόνο			



Πίνακας 2 – Κόστος ανά φάση επεξεργασίας στις σύγχρονες εφαρμογές

## 8.2 Συμπεράσματα

Σύμφωνα με το παράδειγμα που χρησιμοποιήσαμε (κεφάλαιο 3) αλλά και την παραπάνω κατηγοριοποίηση των εφαρμογών των παράλληλων και κατανεμημένων συστημάτων, για την επίτευξη υψηλής αποδοτικότητας αναφορικά με τις φάσεις επεξεργασίας προκύπτουν τα εξής βασικά συμπεράσματα. Βασικό παράγοντα επίτευξης αποδοτικότητας αποτελεί η αποφυγή μεταφοράς του συνόλου των δεδομένων του συστήματος για επεξεργασία στην φάση της συγχώνευσης. Είναι σημαντικό τα δεδομένα να επεξεργάζονται σε τοπικό επίπεδο και να μεταφέρονται μόνο τα αποτελέσματα της επεξεργασίας, μειώνοντας έτσι σε σημαντικό βαθμό τον όγκο δεδομένων που θα πρέπει να μεταφερθούν αλλά και το κόστος της επεξεργασίας τους στην τελική φάση της συγχώνευσης. Όσο μεγαλύτερο είναι το εύρος του δικτύου των συνδεδεμένων κόμβων του κατανεμημένου συστήματος, και ο όγκος των δεδομένων που διαχειρίζεται το σύστημα, τόσο μεγαλύτερο το κόστος διαχείρισής τους σε όλες τις φάσεις επεξεργασίας.

Συνοψίζοντας, επισημαίνεται ότι για την επίτευξη αποτελεσματικότητας, θα πρέπει να οριστεί από τους αλγόριθμους το μέγεθος συμμετοχής της κάθε φάσης επεξεργασίας, σύμφωνα με την αρχιτεκτονική και τα χαρακτηριστικά του συστήματος όπως παρουσιάζονται ακολούθως.

- **Τοπική επεξεργασία**

Αύξηση συμμετοχής της φάσης της τοπικής επεξεργασίας όταν:

- α) τα δεδομένα είναι συγκριτικά πολλά σε σχέση με το μέγεθος του συστήματος.
- β) το σύστημα συνδέεται μέσω δικτύου τηλεπικοινωνιών.
- γ) η απόσταση μεταξύ των κόμβων του συστήματος είναι μεγάλη (γεωγραφικά απομακρυσμένη απόσταση).

- **Μεταφορά δεδομένων**

Αύξηση συμμετοχής της φάσης της μεταφοράς δεδομένων όταν:

- α) τα δεδομένα που πρέπει να αποσταλούν είναι περιορισμένα.
- β) οι κόμβοι του συστήματος συνδέονται μέσω δικτύου τηλεπικοινωνιών.
- γ) η απόσταση μεταξύ των κόμβων είναι μικρή (δηλαδή εντός ενός κτηριακού συγκροτήματος).

- **Συγχώνευση**

Αύξηση συμμετοχής της φάσης της τελικής συγχώνευσης των δεδομένων όταν:

- α) τα δεδομένα είναι συγκριτικά περιορισμένα σε σχέση με το μέγεθος του συστήματος.
- β) οι κόμβοι του συστήματος συνδέονται μέσω καλωδίου (ethernet).
- γ) η απόσταση μεταξύ των κόμβων του συστήματος είναι μικρή (δηλαδή εντός ενός κτηριακού συγκροτήματος).

Στον ακόλουθο πίνακα παρουσιάζεται συνοπτικά η προτιμώμενη συμμετοχή κάθε φάσης επεξεργασίας για τα παράλληλα και κατανεμημένα συστήματα σύμφωνα με τα βασικά τους χαρακτηριστικά/γνωρίσματα για την επίτευξη αποδοτικότητας .

		Δεδομένα/τιμές		Δίκτυο		Απόσταση (μεταξύ των κόμβων)	
		Περιορισμένα	Αυξημένα	Τηλεπ/νίες	Ενσύρματο	Περιορισμένη	Αυξημένη
<b>Φάσεις επεξεργασίας</b>	<b>Τοπική επεξεργασία</b>						
	<b>Μεταφορά δεδομένων</b>						
	<b>Συγχώνευση</b>						

 **Αυξημένη συμμετοχή**

Πίνακας 3 – Εκτίμηση ορισμού συμμετοχής κάθε φάσης επεξεργασίας βάσει των χαρακτηριστικών του

## ***Βιβλιογραφία***

- Βασιλειάδης Π. (2008), *Καταναμημένες βάσεις δεδομένων, Κεφάλαιο 2 διαθέσιμο στο* [http://courses.dbnet.ntua.gr/fsr/3229/2\\_distributed.pdf](http://courses.dbnet.ntua.gr/fsr/3229/2_distributed.pdf).
- Δημακόπουλος Β. (2015), *Παράλληλα συστήματα και Προγραμματισμός*, 1η αναθεωρημένη έκδοση 2017, ΣΕΑΒ.
- Παπαδάκης Σ. & Διαμαντάρας Κ. (2012), *Προγραμματισμός και αρχιτεκτονική συστημάτων παράλληλης επεξεργασίας*, Εκδόσεις Κλειδάριθμος.
- Ajay D. Kshemkalyani, Mukesh Singhal (2008), *Distributed Computing Principles, Algorithms, and Systems*, Cambridge University Press.
- Bharat S. Rawal, Berman Lewis and Ramcharan Harold (July 2013), *Multi-Client/Multi-Server Split Architecture*, <https://www.researchgate.net>.
- Bhardwaj A. (September 2012), *Role of Fragmentation in Distributed database system*, Article in *Parallel Computing Surveys*, <https://www.researchgate.net/publication/323826232>.
- Coulouris G, Dollimore J., Kindberg T, Blair G. (2012), *Distributed Systems, Concepts and Design*, 5<sup>th</sup> Edition, Pearson Education.
- Haron M. (2018), *Query Processing and Optimization in Distributed Database Systems*, article retrieved <https://www.researchgate.net/publication/327917278>
- Hose K., Vlachou A. (2012). *A survey of skyline processing in highly distributed environments*, The Vldb Journal - VLDB. 21. 1-26, 10.1007/s00778-011-0246-6.
- Iacob N. (2010), *Data replication in distributed environments*, Annals - Economy Series, Constantin Brancusi University, Faculty of Economics, vol. 4, pages 193-202
- Jarke M., Koch J. (1984), *Query Optimization in Database Systems*, Computing Surveys, Vol. 16, No 2.
- Kapoor R. (2013), *Cost Estimates & Optimization of Queries Distributed Databases*, International Journal of Engineering Research & Technology (IJERT)ISSN: 2278-0181, Vol. 2 Issue 6.
- Kapoor R., Dr. Virk R.S. (2013), *Selectivity & Cost Estimates in Query Optimization in Distributed Databases*, International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471, Vol. 2 Issue 6, pp: 1-4.

Katambo Kituta Ezechiel, Dr. Ruchi Agarwal, Dr. Baijnath Kaushik (2017), *Synchronous and Asynchronous Replication*, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 3, Issue 7, ISSN: 2456-3307.

Maarten van Steen, Andrew S. Tanenbaum (2016), *A brief introduction to distributed systems*, Computing (2016) 98:967–1009, DOI 10.1007/s00607-016-0508-7, article retrieved [www.Springerlink.com](http://www.Springerlink.com).

Maffeis Olsen S. & Associates (1998), *Client/Server Term Definition*, Encyclopedia of Computer Science, International Thomson Computer Publishing.

Pacheco P.S. (2011), *Εισαγωγή στον παράλληλο προγραμματισμό*, μτφσ. Κωστάκης Δ., επιμ. Γκιζόπουλος Δ., Κλειδάριθμος, 2015.

Raipurkar A., Bammote G. R. (2013), Query Processing in Distributed Database Through Data Distribution, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 2.

Sabu M. Thampi (2009), *Introduction to Distributed Systems*, L.B.S. Institute of Technology for Women, Trivandrum, Kerala, India.

Tamer M. Özsu (2002), *Distributed Database Systems*, article received <https://www.researchgate.net/publication/228806512>.

Tamer M. Ozsu, Valduriez P. (1996), *Distributed and Parallel Database Systems*, ACM Computing Surveys, Vol. 28, No. 1.

Tamer M. Ozsu, Valduriez Patrick (2011), Principles of Distributed Database Systems, 3rd Edition, Springer.

Virendra Dilip Thoke (2015), *Theory of distributed computing and parallel processing with its application, advantages and disadvantages*, International Journal of Innovation in Engineering, Research and Technology [IJERT], ICITDCEME'15 Conference Proceedings, ISSN No - 2394-3696.

<https://www.tutorialcup.com/dbms/data-fragmentation.htm>

<https://www.techopedia.com/definition/26224/query-optimizer>

[https://cs.stanford.edu/people/eroberts/courses/soco/projects/distributed-computing/html/body\\_history.html](https://cs.stanford.edu/people/eroberts/courses/soco/projects/distributed-computing/html/body_history.html)

<https://www.milesight.com/support/troubleshooting/NVR-P2P-on-M-Sight-Pro>



[https://el.wikipedia.org/wiki/Παράλληλα\\_και\\_κατανεμημένα\\_συστήματα](https://el.wikipedia.org/wiki/Παράλληλα_και_κατανεμημένα_συστήματα)

<https://www.c-sharpcorner.com/UploadFile/f3068a/sql-server-query-optimization-day-2-no-of-joins-does-mat>

[https://datacadamia.com/\\_detail/data/concurrency/taxonomy\\_of\\_parallel\\_architecture.jpg?id=data%3Aconcurrency%3Aarchitecture](https://datacadamia.com/_detail/data/concurrency/taxonomy_of_parallel_architecture.jpg?id=data%3Aconcurrency%3Aarchitecture)

<https://www.codeit-project.eu/el/algorithmic-design-data-structures>