



# Εισαγωγή



Για τη δημιουργία των διαφανειών έχει χρησιμοποιηθεί υλικό από τις διαφάνειες παραδόσεων που βασίζονται στο βιβλίο,

**Silberschatz, Galvin and Gagne, "Operating Systems Concepts", 6<sup>th</sup> Edition.**

Οι διαφάνειες αυτές βρίσκονται στο δικτυακό τόπο:

<http://www.cs.purdue.edu/homes/yau/cs503/>

# Προτεινόμενη Βιβλιογραφία

- Distributed Systems, Andrew Tanenbaum & Maarten van Steen, Prentice Hall, 2002
- Introduction to Distributed Algorithms, Gerard Tel, Cambridge University Press, 1994
- Distributed Algorithms, Nancy Lynch, Morgan Kaufman, 1996
- Distributed Systems: Concepts and Design, G. Coulouris, J. Dollimore and T. Kindberg, Addison-Wesley, 3<sup>rd</sup> edition, 2001



# Εισαγωγή

- Ιστορική Εξέλιξη
- Βασικές έννοιες
- Στόχοι
- Ζητήματα Σχεδίασης
- Ζητήματα Υλικού
- Ζητήματα Λογισμικού

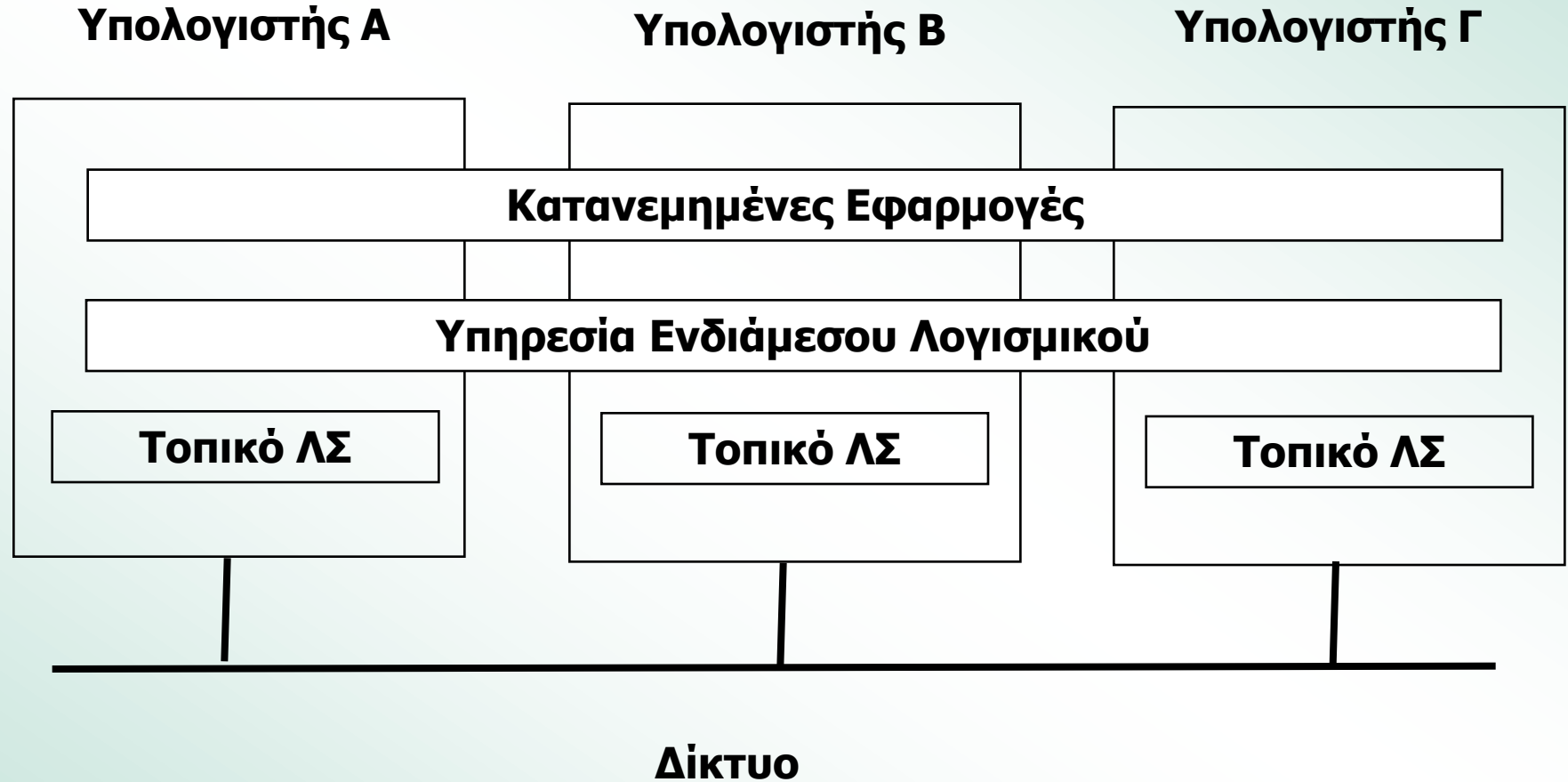
# Ιστορική Εξέλιξη

- Η ιστορία των κατανεμημένων συστημάτων ξεκινά από τα μέσα της δεκαετίας του 1980.
- Αιτίες της εξέλιξης αυτής είναι:
  - Η ανάπτυξη ισχυρών μικροεπεξεργαστών
  - Η εφεύρεση των δικτύων υπολογιστών υψηλής ταχύτητας:
    - Τοπικά δίκτυα (LANs)
    - Δίκτυα ευρείας περιοχής (WANs)

# Ορισμός Κατανεμημένου Συστήματος

- «Είναι μια συλλογή από ανεξάρτητους υπολογιστές, οι οποίοι εμφανίζονται στους χρήστες τους ως ένα ενιαίο συνεκτικό σύστημα»
- Παρατηρήσεις:
  - Υλικό: αυτονομία υπολογιστών
  - Λογισμικό: εικόνα ενιαίου συστήματος με χρήση ενδιάμεσου λογισμικού

# Ενδιάμεσο λογισμικό



# «Ενδιάμεσο» Λογισμικό (Middleware)

- Πρόκειται για λογισμικό που στόχο έχει να κρύψει την ετερογένεια και να παρέχει ένα εύχρηστο μοντέλο για προγραμματισμό
- Είναι ένα σύνολο διεργασιών ή αντικειμένων σε ένα κατανεμημένο σύστημα που αλληλεπιδρούν για να υποστηρίξουν την επικοινωνία και το διαμοιρασμό πόρων
- Υποστηρίζει αφαιρέσεις όπως απομακρυσμένη επίκληση διεργασίας (Remote Method Invocation – RMI), επικοινωνία μιας ομάδας διεργασιών, αντιγραφή κοινόχρηστων δεδομένων και μετάδοση πολυμεσικών δεδομένων σε πραγματικό χρόνο



# «Ενδιάμεσο» Λογισμικό (συνέχεια)

- SUN's RPC (Remote Procedure Call)
- Common Object Request Broker Architecture (CORBA)
- Java Remote Method Invocation (RMI)
- Microsoft's DCOM
- Η υποστήριξη ενδιάμεσου λογισμικού έχει διευκολύνει τον προγραμματισμό των κατανεμημένων συστημάτων, ορισμένα όμως ζητήματα απαιτούν υποστήριξη στο επίπεδο εφαρμογών

# Παραδείγματα Κατανεμημένων Συστημάτων

- Μια αίθουσα υπολογιστών που περιλαμβάνει μια ομάδα επεξεργαστών που εκχωρούνται δυναμικά όπου χρειάζονται και όχι σε συγκεκριμένους χρήστες
- Ένα σύστημα ροής πληροφοριών ροής εργασιών που υποστηρίζει αυτόματη διεκπεραίωση παραγγελιών
- WWW
- Το Internet
- Τα Intranets
- Κινητός και «Πανταχού Παρών» Υπολογισμός (Mobile και Ubiquitous Computing)

# To Internet

- Το Internet είναι ένα μεγάλο κατανεμημένο σύστημα. Επιτρέπει στους χρήστες, οπουδήποτε και αν βρίσκονται, να χρησιμοποιούν υπηρεσίες όπως το Web, το e-mail και η μεταφορά αρχείων
- Το σύνολο των υπηρεσιών αυτών είναι ανοικτό (open-ended)
- Η υλοποίηση του Internet και οι υπηρεσίες που υποστηρίζει συνεπάγονται την υλοποίηση πρακτικών λύσεων σε πολλά ζητήματα κατανεμημένων συστημάτων (DNS, client-server computing)

# Intranets

- Πρόκειται για ένα «κομμάτι» του Internet που διοικείται από κάποιο συγκεκριμένο φορέα, ο οποίος επιβάλλει τοπικές πολιτικές ασφάλειας
- Το Intranet χρησιμοποιεί την οικογένεια πρωτοκόλλων TCP/IP και τις υπηρεσίες αυτής (web, mail) πιθανά σε συνδυασμό με κάποιο Λειτουργικό Σύστημα Δικτύου
- Επιτρέπει την ελεγχόμενη πρόσβαση στους πόρους του
- Συνίσταται κυρίως στα ακόλουθα δομικά στοιχεία: υπηρεσίες αρχείων για το διαμοιρασμό δεδομένων, firewalls, κόστος λογισμικού και υποστήριξης

# Κινητός και «Πανταχού Παρών» Υπολογισμός (Mobile and Ubiquitous Computing)

- Οι τεχνολογικές εξελίξεις στη σμίκρυνση των συσκευών και στην ασύρματη δικτύωση, έχουν οδηγήσει στην ολοκλήρωση μικρών και φορητών υπολογιστικών συσκευών σε κατανεμημένα συστήματα
- Τέτοιες συσκευές είναι οι φορητοί υπολογιστές, οι συσκευές χειρός (PDAs, κινητά τηλέφωνα, κάμερες), τα έξυπνα ρολόγια (smart watches) και διάφορες συσκευές ενσωματωμένες σε οικιακές συσκευές

# Γιατί Κατανεμημένα Συστήματα;

- Διαμοιρασμός πόρων
  - Διαμοιρασμός και εκτύπωση αρχείων σε απομακρυσμένα μέρη
  - Επεξεργασία πληροφορίας σε κατανεμημένες βάσεις δεδομένων
  - Χρήση εξειδικευμένων απομακρυσμένων συσκευών υλικού
- Επιτάχυνση των υπολογισμών– *καταμερισμός φόρτου load sharing*
- Αξιοπιστία (Reliability) – ανίχνευση και ανάνηψη (detect and recover) από αστοχία δικτυακού τόπου, μεταφορά κώδικα, και επανένταξη προβληματικού δικτυακού τόπου
- Επικοινωνία – πέρασμα μηνυμάτων (message passing)
- Ευκολία επέκτασης ή προσαρμογής της κλίμακας του μεγέθους τους

# Ζητήματα Σχεδίασης

- Διαφάνεια (Transparent)
- Ανοικτή Λειτουργία
- Ευελιξία (Flexibility)
- Αξιοπιστία (Reliability)
  - Διαθεσιμότητα (Availability)
  - Ανεκτικότητα σε Σφάλματα (Fault Tolerance)
- Απόδοση (Performance)
- Ικανότητα Κλιμάκωσης (Scalability)

# Διαφάνεια (Transparency)

- Διαφάνεια Τοποθέτησης: οι χρήστες δε γνωρίζουν που βρίσκονται τοποθετημένοι οι πόροι του υλικού και του λογισμικού
- Διαφάνεια Μετανάστευσης: οι πόροι είναι ελεύθεροι να μετακινούνται από μια τοποθεσία σε μια άλλη, χωρίς να χρειάζεται να αλλάξουν τα ονόματά τους, ακόμη κι όταν μετακινούνται
- Διαφάνεια Ομοιοτυπίας: οι χρήστες δεν γνωρίζουν πόσα ομοιότυπα υπάρχουν
- Διαφάνεια Ταυτοχρονισμού: οι χρήστες δεν γνωρίζουν αν ένας πόρος είναι κοινόχρηστος
- Διαφάνεια Αστοχίας: οι χρήστες δεν γνωρίζουν περί της αστοχίας ή της ανάκαμψης ενός πόρου
- Διαφάνεια προσπέλασης: οι χρήστες δεν γνωρίζουν την ύπαρξη διαφορών στην αναπαράσταση δεδομένων και την προσπέλαση ενός πόρου



# Ανοικτή Λειτουργία

- Ανοικτό Κατανεμημένο Σύστημα είναι ένα σύστημα το οποίο παρέχει υπηρεσίες σύμφωνα με κάποιους καθιερωμένους κανόνες, οι οποίοι περιγράφουν τη σύνταξη και τη σημειολογία των υπηρεσιών αυτών.
- Οι υπηρεσίες καθορίζονται μέσω interfaces, που περιγράφονται σε κάποια γλώσσα IDL.
- IDL: καθορίζει μόνο τη σύνταξη (ονόματα συναρτήσεων, παραμέτρους, επιστρεφόμενες τιμές, εξαιρέσεις, κλπ.).
- Η σημειολογία καθορίζεται συνήθως σε φυσική γλώσσα.

# Ιδιότητες Διασυνδέσεων Ανοικτής Λειτουργίας

- Οι διασυνδέσεις (interfaces) πρέπει να είναι πλήρεις και ουδέτερες.
- Διαλειτουργικότητα: καθορίζει κατά πόσο δύο διαφορετικές υλοποιήσεις διαφορετικών κατασκευαστών μπορούν να συνυπάρξουν και να συνεργάζονται με βάση κοινά πρότυπα.
- Φορητότητα
- Ευελιξία: εύκολη διευθέτηση με χρήση διαφορετικών συστατικών στοιχείων, ενδεχομένως από διαφορετικούς κατασκευαστές
- Επεκτασιμότητα: εύκολη προσθήκη νέων στοιχείων ή αντικατάσταση υπάρχοντων στοιχείων, χωρίς να επηρεάζονται αυτά που παραμένουν.

# Ευελιξία (Flexibility)

- Μονολιθικός πυρήνας (monolithic kernel)
- Μικροπυρήνας (Microkernel) – Μεταφορά αρμοδιοτήτων πυρήνα σε εξυπηρετητές που τρέχουν σε κατάσταση χρήστη (δεν παρέχει υπηρεσίες συστήματος αρχείων, πλήρους διαχείρισης διεργασιών και διαχείριση όλων των κλήσεων συστήματος)

# Αξιοπιστία (Reliability)

- Όταν καταρρέει μια μηχανή κάποια άλλη πρέπει να αναλάβει την εργασία της. Υπάρχουν διάφορες όψεις της αξιοπιστίας: διαθεσιμότητα (availability), και ανεκτικότητα σε σφάλματα (fault tolerance)
- Η τήρηση ομοιοτύπων πόρων (όπως π.χ. ομοιοτύπων αρχείων) που ονομάζεται πλεονασμός (redundancy) αυξάνει τη διαθεσιμότητα αλλά δημιουργεί προβλήματα συνέπειας
- Ασφάλεια από μη εξουσιοδοτημένη χρήση

# Απόδοση (Performance)

- Χρησιμοποιούνται πολλοί τρόποι για τη μέτρηση της απόδοσης: ο χρόνος απόκρισης (response time), ο ρυθμός απόδοσης (throughput), ο βαθμός αξιοποίησης του συστήματος (utilization), η χρησιμοποιούμενη χωρητικότητα του δικτύου (capacity)
- Το δίκτυο απαραίτητο αλλά και πηγή καθυστερήσεων
- Εργασίες που επιδεικνύουν χαμηλής τάξης παραλληλισμό (fine-grained parallelism) και εργασίες με παραλληλισμό υψηλής τάξης (coarse-grain parallelism)

# Ικανότητα Κλιμάκωσης (Scalability)

- Μετριέται τουλάχιστον σε τρεις διαστάσεις:
  - Αύξηση του μεγέθους
  - Γεωγραφική επεκτασιμότητα
  - Διαχειριστική επέκταση
- Προβλήματα επεκτασιμότητας:
  - Αύξηση μεγέθους: δημιουργεί συμφόρηση σε συγκεντρωτικές υπηρεσίες
  - Μη συγκεντρωτικές υπηρεσίες: δημιουργούν προβλήματα ασφάλειας
  - Μη συγκεντρωτικά δεδομένα: δημιουργούν προβλήματα απόδοσης

# Ικανότητα Κλιμάκωσης (Scalability) (2)

- Θα μπορέσουν οι μέθοδοι που αναπτύσσονται να εξασφαλίσουν την ικανότητα κλιμάκωσης σε μεγάλα κατανεμημένα συστήματα;
- Όχι κεντρικοποιημένα δομικά τμήματα, πίνακες και αλγόριθμοι
- Θα πρέπει να χρησιμοποιηθούν κατανεμημένοι αλγόριθμοι με τα παρακάτω χαρακτηριστικά:
  - Καμία μηχανή δεν έχει ολοκληρωμένη πληροφόρηση για την κατάσταση του συστήματος
  - Οι μηχανές παίρνουν αποφάσεις βασιζόμενες μόνο στις τοπικά διαθέσιμες πληροφορίες
  - Η κατάρρευση μιας μηχανής δεν καταστρέφει τον αλγόριθμο
  - Δεν υπάρχει καθολικό ρολόι

# Ικανότητα Κλιμάκωσης (Scalability) (3)

## — Γεωγραφική επεκτασιμότητα:

- παρουσιάζονται προβλήματα εφαρμογής σύγχρονης επικοινωνίας σε δίκτυα ευρείας περιοχής
- Στα ευρείας περιοχής δίκτυα η επικοινωνία είναι τύπου point to point, δηλαδή εγγενώς αναξιόπιστη
- Αντιμετωπίζει προβλήματα με συγκεντρωτικές υπηρεσίες

## — Διαχειριστική επεκτασιμότητα:

- Δημιουργεί προβλήματα χρήσης, διαχείρισης και ασφάλειας των πόρων



# Ικανότητα Κλιμάκωσης (Scalability) (4)

## — Τεχνικές Επέκτασης

- Απόκρυψη καθυστέρησης επικοινωνίας μέσω ασύγχρονης κλήσης: π.χ. σε μια πολυνηματική διεργασία το πρόγραμμα του νήματος που εκτελεί την απομακρυσμένη κλήση μπλοκάρεται, ενώ τα άλλα νήματα συνεχίζουν κανονικά τη λειτουργία τους.
- Κατανομή: ένα τμήμα διαιρείται σε μικρότερα τμήματα (π.χ. DNS)
- Αναπαραγωγή (replication): ύπαρξη πολλαπλών αντιγράφων στοιχείων σε ένα κατανεμημένο σύστημα.
  - Δημιουργούνται προβλήματα συνέπειας αντιγράφων

# Κατηγοριοποίηση Υπολογιστικών Συστημάτων κατά Flynn

- Η κατηγοριοποίηση των συστημάτων γίνεται με βάση δύο κριτήρια:
  - Το πλήθος των ροών εντολών (instruction streams)
  - Το πλήθος των ροών δεδομένων (data streams)
- Συστήματα τύπου SISD (Simple Instruction Simple Data), π.χ. PCs
- Συστήματα τύπου SIMD (Simple Instruction Multiple Data), αναφέρεται σε επεξεργαστές σε παράθεση (array processors), π.χ. υπερυπολογιστές (supercomputers)

# Κατηγοριοποίηση Υπολογιστικών Συστημάτων κατά Flynn (συνέχεια)

- Συστήματα τύπου MISD (Multiple Instruction Simple Data), δεν υπάρχουν γνωστοί υπολογιστές στην κατηγορία αυτή
- Συστήματα τύπου MIMD (Multiple Instruction Multiple Data), π.χ. κατανεμημένα συστήματα (distributed systems)
- Η κατηγοριοποίηση αυτή δεν μας καλύπτει για την περαιτέρω κατηγοριοποίηση των συστημάτων τύπου MIMD

# Ζητήματα Υλικού

- Με ποιον τρόπο διασυνδέονται και επικοινωνούν οι πολλαπλές ΚΜΕ σε ένα ΚΣ;
- Πολυεπεξεργαστές (multiprocessors) με διαμοιραζόμενη μνήμη και πολυυπολογιστές (multicomputers π.χ. ένα τοπικό δίκτυο)
- Σε ένα πολυεπεξεργαστή υπάρχει ένας και μοναδικός (ιδεατός) χώρος διευθύνσεων
- Αρχιτεκτονική του δικτύου διασύνδεσης (αρτηρίας – bus και μεταγωγής – switched)

# Ζητήματα Υλικού (συνέχεια)

- Άρα δημιουργούνται τέσσερις κατηγορίες: οι πολυεπεξεργαστές αρτηρίας (bus multiprocessors), οι πολυεπεξεργαστές μεταγωγής (switched multiprocessors), οι πολυυπολογιστές αρτηρίας (bus multicomputers), οι πολυυπολογιστές μεταγωγής (switched multicomputers)

# Ζητήματα Υλικού (συνέχεια)

- Οι πολυεπεξεργαστές είναι στενά συνδεδεμένα συστήματα, ενώ οι πολυυπολογιστές είναι χαλαρά συνδεδεμένα συστήματα
- Στενά συνδεδεμένα συστήματα (tightly coupled)
  - Υψηλός ρυθμός μετάδοσης δεδομένων
  - Μικρή καθυστέρηση μηνύματος
- Χαλαρά συνδεδεμένα συστήματα (loosely coupled)
  - Χαμηλός ρυθμός μετάδοσης δεδομένων
  - Μεγάλη καθυστέρηση μηνύματος

# Πολυεπεξεργαστές Αρτηρίας

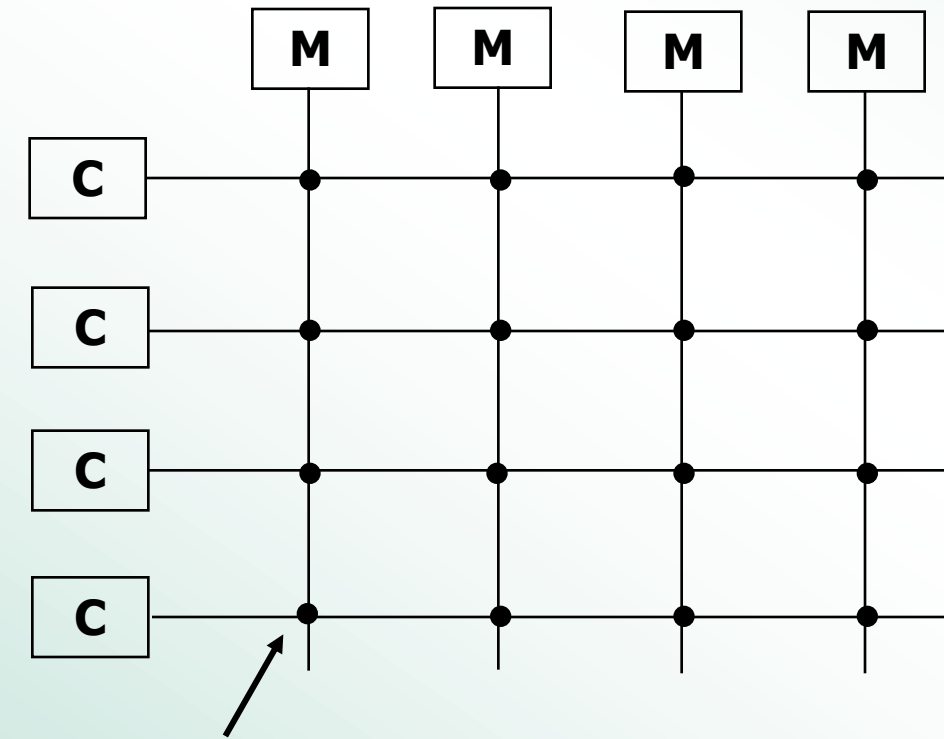
- Αποτελούνται από έναν αριθμό ΚΜΕ, όπου όλες συνδέονται σε μια κοινή αρτηρία και ένα τμήμα μνήμης
- Η αύξηση του πλήθους των ΚΜΕ συνεπάγεται μείωση της απόδοσης του συστήματος
- Λύση: κρυφή μνήμη μεταξύ ΚΜΕ και αρτηρίας
- Στην κρυφή μνήμη τηρείται αντίγραφο των πιο πρόσφατων προσπελάσεων
- Όλες οι αιτήσεις προς τη μνήμη περνάνε πρώτα από την κρυφή μνήμη
- Όσο μεγαλύτερο το ποσοστό επιτυχίας, τόσο μεγαλύτερη δυνατότητα σύνδεσης περισσότερων ΚΜΕ στην αρτηρία

# Πολυεπεξεργαστές Αρτηρίας (συνέχεια)

- Αλλά: **Πρόβλημα συνέπειας της μνήμης!**
- Τι συμβαίνει στην περίπτωση που κάποια ΚΜΕ αλλάξει την τιμή κάποιας θέσης μνήμης;
- Όταν μια λέξη εγγράφεται στην κρυφή μνήμη, εγγράφεται ταυτόχρονα και στην κύρια μνήμη (write-through cache)
- Όλες οι κρυφές μνήμες παρακολουθούν συνεχώς την αρτηρία και όταν εντοπίσει εγγραφή είτε διαγράφει τη δική της θέση, ή την ενημερώνει με τη νέα τιμή (snoopy cache)

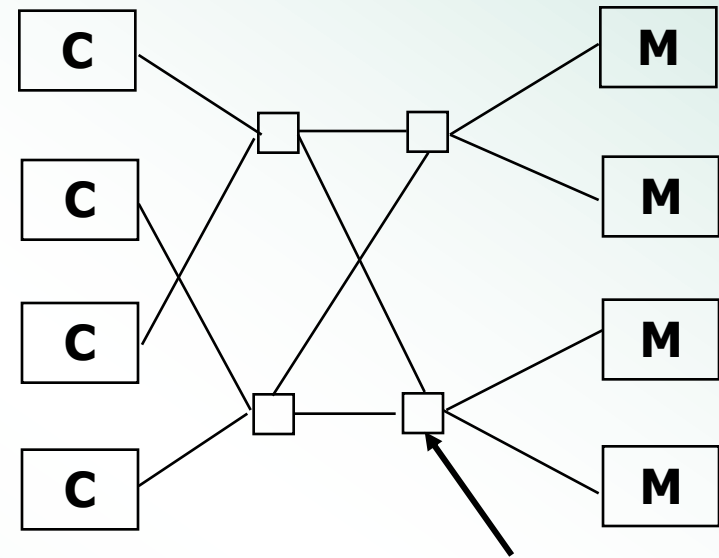


# Πολυεπεξεργαστές Μεταγωγής



σημείο μεταγωγής

A) Πίνακας μεταγωγής



Μεταγωγή 2Χ2

B) Δίκτυο ωμέγα

# Πολυεπεξεργαστές Μεταγωγής (συνέχεια)

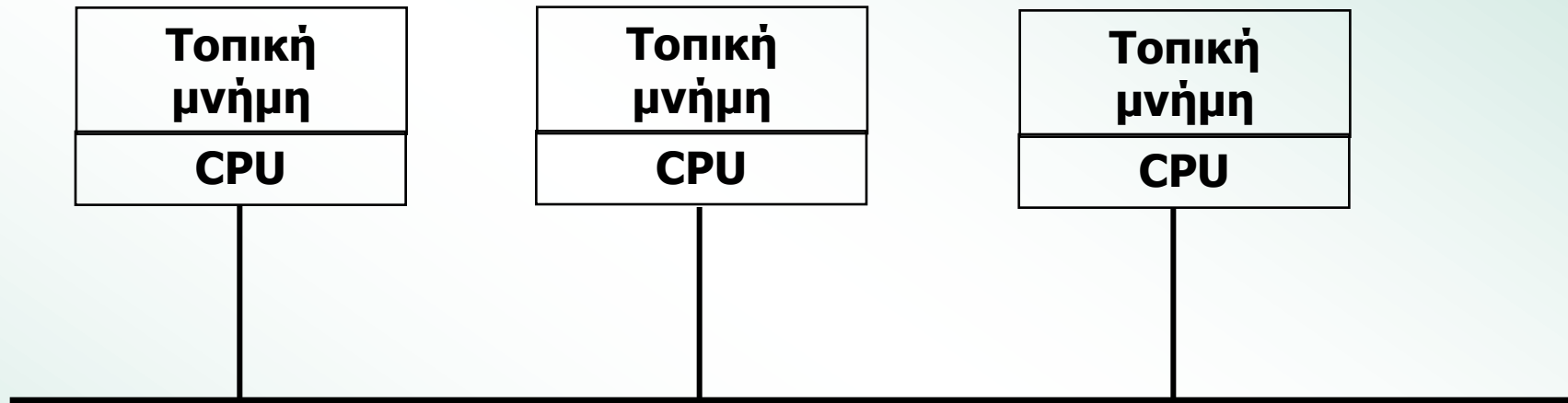
- Μειονέκτημα πίνακα μεταγωγής : αν έχουμε  $n$  CPUs και  $n$  μνήμες, τότε χρειάζονται  $n^2$  σημεία μεταγωγής, που είναι απαγορευτικό για μεγάλα  $n$ .
- Μειονέκτημα δικτύου ωμέγα : αν έχουμε  $n$  CPUs και  $n$  μνήμες, τότε χρειάζονται  $n$  στάδια μεταγωγής, που το καθένα περιέχει  $\log_2 n$  μεταγωγούς.
- Λύση: η κατασκευή μηχανών NUMA (NonUniform Memory Access – Μη Ομοιόμορφη Πρόσβαση Μνήμης)

# Πολυεπεξεργαστές Μεταγωγής (συνέχεια)

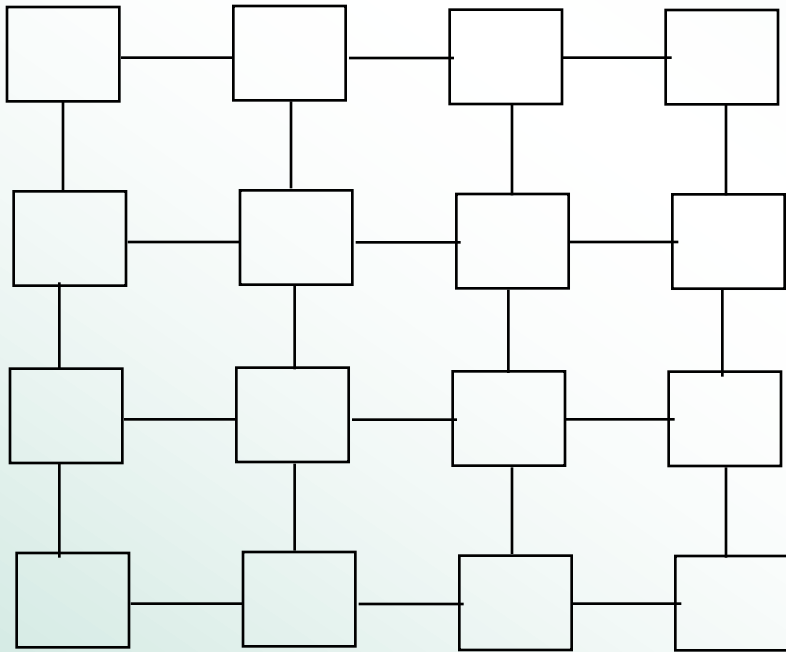
## — Μηχανές NUMA:

- Μία μνήμη συσχετίζεται με κάθε CPU
- Κάθε CPU έχει γρήγορη πρόσβαση στη δική της μνήμη, αλλά αργή πρόσβαση σε όλες τις άλλες μνήμες
- Πρόβλημα: πώς θα γίνει η τοποθέτηση του προγράμματος και των δεδομένων ώστε να οι περισσότερες προσπελάσεις να γίνονται στην τοπική μνήμη;

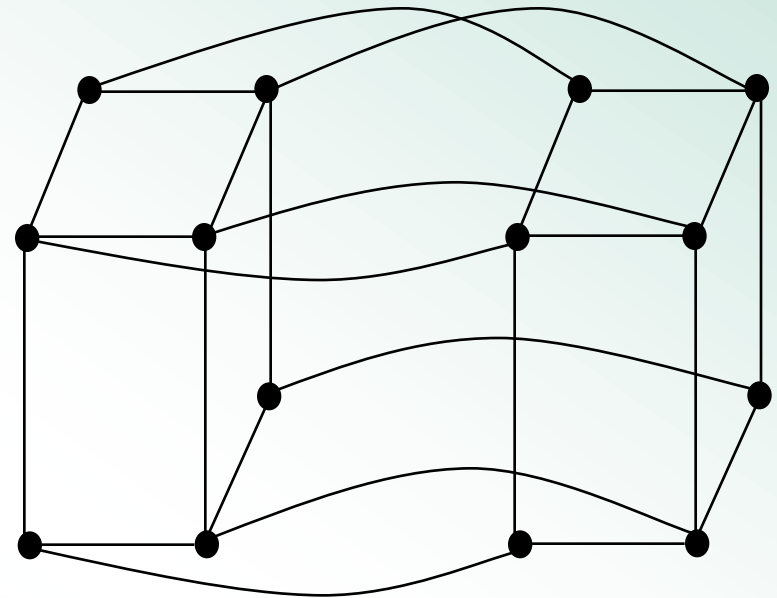
# Πολυυπολογιστές Αρτηρίας



# Πολυυπολογιστές Μεταγωγής



πλέγμα



Υπερκύβος 3-Δ

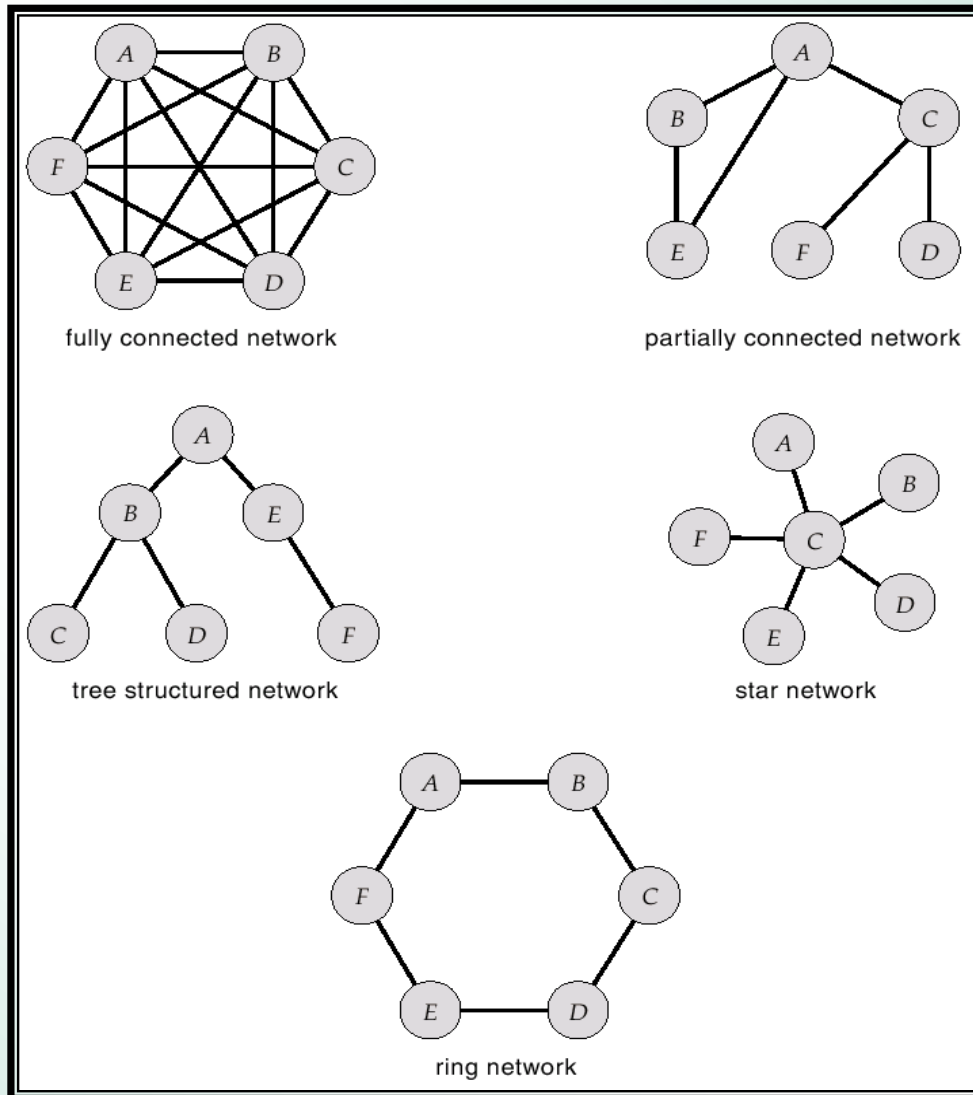
# Ομοιογενείς και Ετερογενείς Πολυυπολογιστές

- Ομοιογενείς πολυυπολογιστές: υπάρχει ουσιαστικά ένα και μόνο δίκτυο διασύνδεσης που χρησιμοποιεί παντού την ίδια τεχνολογία. Όλοι οι επεξεργαστές είναι ίδιοι και γενικά έχουν πρόσβαση στην ίδια ποσότητα ιδιωτικής μνήμης. Χρησιμοποιούνται περισσότερο ως παράλληλα συστήματα.
- Ετερογενείς πολυυπολογιστές: υπάρχει ετερογένεια στο δίκτυο διασύνδεσης και τους επεξεργαστές

# Τοπολογία

- **Οι τόποι σε ένα καταναμημένο σύστημα μπορούν να συνδέονται με μια ποικιλία τρόπων, σύμφωνα με τα ακόλουθα κριτήρια:**
  - **Κόστος.** Πόσο κοστίζει να συνδέσουμε τους διάφορους τόπους στο σύστημα;
  - **Κόστος Επικοινωνίας.** Πόσος χρόνος απαιτείται να στείλουμε ένα μήνυμα από τον τόπο A στον τόπο B;
  - **Αξιοπιστία (Reliability).** Αν αστοχήσει ένας σύνδεσμος ή ένας τόπος του συστήματος, μπορούν οι υπόλοιποι τόποι να επικοινωνήσουν μεταξύ τους
- **Οι διάφορες τοπολογίες αναπαρίστανται ως γράφοι των οποίων οι κόμβοι αντιστοιχούν σε τόπους. Μια ακμή από τον κόμβο A στον κόμβο B αντιστοιχεί σε άμεση σύνδεση μεταξύ των δύο τόπων**

# Τοπολογία (συνέχεια)





# Ζητήματα Λογισμικού

- Το λογισμικό πιο σημαντικό από το υλικό στα ΚΣ
- Η εικόνα που ένα σύστημα παρουσιάζει στους χρήστες του εξαρτάται από το λογισμικό του ΛΣ και όχι από το υλικό
- Χαλαρά και Στενά συνδεδεμένο λογισμικό: ανάλογα με το βαθμό ανεξαρτησίας και αλληλεπίδρασης χρηστών και μηχανών
- Ένα λειτουργικό σύστημα στενής σύνδεσης χαρακτηρίζεται γενικά ως κατανεμημένο λειτουργικό σύστημα και χρησιμοποιείται για τη διαχείριση πολυεπεξεργαστών και ομοιογενών πολυυπολογιστών
- Ένα λειτουργικό σύστημα χαλαρής σύνδεσης χαρακτηρίζεται γενικά ως δικτυακό λειτουργικό σύστημα και χρησιμοποιείται για τη διαχείριση ετερογενών πολυυπολογιστών

# Λειτουργικά Συστήματα Δικτύου – Network Operating Systems

- Οι χρήστες είναι ενήμεροι για την πολλαπλότητα των μηχανών και τη θέση τους σε ένα δίκτυο. Η πρόσβαση στους διάφορους πόρους του δικτύου γίνεται σαφώς με το να:
  - συνδεόμαστε με τον κατάλληλο απομακρυσμένο υπολογιστή (Remote logging)
  - μεταφέρουμε δεδομένα από απομακρυσμένες σε τοπικές μηχανές, και αντίστροφα, μέσω του πρωτοκόλλου μεταφοράς αρχείων (File Transfer Protocol (FTP))
  - αντιγράφουμε δεδομένα από απομακρυσμένες σε τοπικές μηχανές, και αντίστροφα, μέσω της εντολής rcp

# Δικτυακά Συστήματα Αρχείων (Network File Systems – NFS)

- Ας θεωρήσουμε το παράδειγμα ενός Τοπικού Δικτύου με χαλαρά συνδεδεμένο υλικό
- Ένα Λειτουργικό Σύστημα Δικτύου προϋποθέτει τη γνώση εκ μέρους του χρήστη για τους πόρους που το δίκτυο διαθέτει
- Μια άλλη προσέγγιση είναι να παρέχεται ένα καθολικό και διαμοιραζόμενο σύστημα αρχείων (file system) προσβάσιμο από όλους τους σταθμούς εργασίας
- Υποστηρίζεται από έναν ή περισσότερους εξυπηρετητές αρχείων (file servers) οι οποίοι εξυπηρετούν αιτήσεις

# Δικτυακά Συστήματα Αρχείων (συνέχεια)

- Οι εξυπηρετητές αρχείων διατηρούν ιεραρχικά συστήματα αρχείων
- Οι πελάτες μπορούν να προσαρτήσουν (mount) μέρος των συστημάτων αρχείων προσαυξάνοντας τα δικά τους τοπικά συστήματα
- Η πρόσβαση στα αρχεία γίνεται ως τα αρχεία αυτά να ήταν τοπικά στο σύστημα αρχείων του πελάτη
- Όλες οι μηχανές μπορούν να τρέχουν το ίδιο ή όχι ΛΣ. Αν όχι πρέπει να συμφωνήσουν σε ένα κοινό σύνολο από μηνύματα

# Δικτυακά Συστήματα Αρχείων (συνέχεια)

- Πρόκειται ουσιαστικά για έναν λειτουργικό σύστημα δικτύου, όπου υπάρχει αυτονομία στις μηχανές και υπάρχουν λίγες απαιτήσεις για το σύνολο του συστήματος (π.χ. NFS της Sun)
- Θα παρουσιάσουμε το NFS (ένα εμπορικά διαθέσιμο προϊόν) ώστε να το συγκρίνουμε με τα πραγματικά κατανομημένα συστήματα που θα παρουσιάσουμε αργότερα
- Το NFS σχεδιάστηκε αρχικά για το Unix, σήμερα όμως το υποστηρίζουν πολλοί κατασκευαστές.

# Δικτυακά Συστήματα Αρχείων (συνέχεια)

- Θα παρουσιάσουμε λίγα πράγματα για την αρχιτεκτονική του, το πρωτόκολλο και την υλοποίηση
- Επιτρέπει σε μια αυθαίρετη συλλογή από πελάτες και εξυπηρετητές να διαμοιράζονται ένα κοινό σύστημα αρχείων πάνω από ένα τοπικό (και όχι μόνο) δίκτυο
- Κάθε εξυπηρετητής εξάγει έναν ή περισσότερους καταλόγους του (/etc/exports)
- Οι εξυπηρετούμενοι προσαρτούν καταλόγους που εξάγονται. Πολλοί σταθμοί εργασίας της Sun δε διέθεταν καν δίσκο (diskless workstations)

# Δικτυακά Συστήματα Αρχείων (συνέχεια)

- Δύο εξυπηρετούμενοι που προσαρτούν τον ίδιο κατάλογο μπορούν να επικοινωνούν μέσω των κοινών αρχείων
- Λόγω του ότι στόχος του NFS είναι η υποστήριξη ενός ετερογενούς συστήματος, με τους πελάτες και τους εξυπηρετητές να τρέχουν πιθανά διαφορετικά ΛΣ σε διαφορετικές πλατφόρμες είναι απαραίτητη η ύπαρξη ενός **πρωτοκόλλου**
- Το NFS χρησιμοποιεί δύο πρωτόκολλα: το πρώτο χειρίζεται την προσάρτηση και το δεύτερο στην πρόσβαση καταλόγων και αρχείων

# Πρωτόκολλο Προσάρτησης

- Ένας πελάτης αποστέλλει ένα όνομα διαδρομής και ζητάει και να προσαρτήσει αυτόν τον κατάλογο
- Ο εξυπηρετητής ελέγχει αν ο κατάλογος έχει εξαχθεί και απαντάει στον πελάτη με ένα χειριστή αρχείου (file handle). Όλες οι επόμενες αναφορές σε αρχεία του προσαρτημένου καταλόγου χρησιμοποιούν τον προσδιοριστή αρχείου
- Πρόγραμμα φλοιού για προσάρτηση
- Αυτόματη προσάρτηση (automounting). Ένα σύνολο από εξυπηρετητές συσχετίζεται με έναν τοπικό κατάλογο. Η αίτηση απευθύνεται σε όλους του εξυπηρετητές και επιλέγεται ο πρώτος που θα απαντήσει – Ανεκτικότητα σε σφάλματα και βελτίωση της απόδοσης



# Πρωτόκολλο Πρόσβασης

- Πρωτόκολλο για τη διαχείριση καταλόγων, την ανάγνωση, εγγραφή και τα χαρακτηριστικά των αρχείων
- Δεν υποστηρίζονται οι κλήσεις συστήματος `open()` και `close()`, αλλά το κάθε μήνυμα περιέχει όση πληροφορία χρειάζεται. Έτσι δεν τηρείται πληροφορία για τα ανοιχτά αρχεία στον εξυπηρετητή (stateless)
- Το NFS χρησιμοποιεί το μηχανισμό προστασίας του Unix

# Κατανεμημένα Λειτουργικά Συστήματα

- Οι χρήστες δεν είναι ενήμεροι για την πολλαπλότητα των μηχανών σε ένα δίκτυο. Η πρόσβαση σε απομακρυσμένους πόρους παρόμοια με την πρόσβαση σε τοπικούς πόρους
- Μετανάστευση Δεδομένων (Data Migration) – μεταφορά δεδομένων με τη μεταφορά ολόκληρου του αρχείου, ή μεταφέροντας μόνο αυτά τα τμήματα του αρχείου που είναι απαραίτητα για την άμεση εργασία
- Μετανάστευση Υπολογισμών (Computation Migration) – μεταφορά του υπολογισμού, και όχι των δεδομένων, μέσα στο κατανεμημένο σύστημα

# Κατανεμημένα Λειτουργικά Συστήματα (συνέχεια)

- Μετανάστευση Διεργασίας (Process Migration) – εκτέλεση μιας ολόκληρης διεργασίας, ή τμημάτων αυτής, σε διαφορετικούς τόπους
  - Καταμερισμός φόρτου – κατανομή των διεργασιών διαμέσου του δικτύου για την εξισορρόπηση του φόρτου εργασίας
  - Επιτάχυνση Υπολογισμών (Computation speedup) – τμήματα της διεργασίας μπορούν να τρέχουν ταυτόχρονα σε διαφορετικούς τόπους
  - Προτίμηση υλικού – η εκτέλεση της διεργασίας μπορεί να απαιτεί εξειδικευμένο υλικό
  - Προτίμηση λογισμικού – το απαιτούμενο λογισμικό μπορεί να είναι διαθέσιμο μόνο σε κάποιο τόπο
  - Πρόσβαση δεδομένων – απομακρυσμένη εκτέλεση διεργασιών, παρά μεταφορά όλων των δεδομένων τοπικά

# Χαρακτηριστικά Κατανεμημένων Συστημάτων

- Πρέπει να υπάρχει μοναδικός και καθολικός μηχανισμός επικοινωνίας των διεργασιών
- Πρέπει να υπάρχει ένα καθολικό σχήμα προστασίας
- Ενιαία διαχείριση των διεργασιών (ένα μοναδικό σύνολο κλήσεων συστήματος που έχουν νόημα στα πλαίσια ενός ΚΣ, διαθέσιμο σε όλες τις μηχανές)
- Άρα στις ΚΜΕ του συστήματος θα πρέπει να εκτελούνται ταυτόσημοι πυρήνες. Ευκολότερος ο συντονισμός δραστηριοτήτων που θα πρέπει να έχουν καθολικό χαρακτήρα (ο κάθε πυρήνας διατηρεί αυτονομία)

# Πλεονεκτήματα Κατανεμημένων έναντι Κεντριοποιημένων Συστημάτων

- Καλύτερος λόγος κόστους/απόδοση
  - (Κατανεμημένα και Παράλληλα Συστήματα)
- Ταχύτητα
- Πολλές εφαρμογές είναι κατανεμημένες από τη φύση τους
- Μεγαλύτερη αξιοπιστία (reliability) λόγω κατανομής του φόρτου εργασίας σε πολλές μηχανές
- Εύκολη κλιμάκωση υπολογιστικής ισχύος

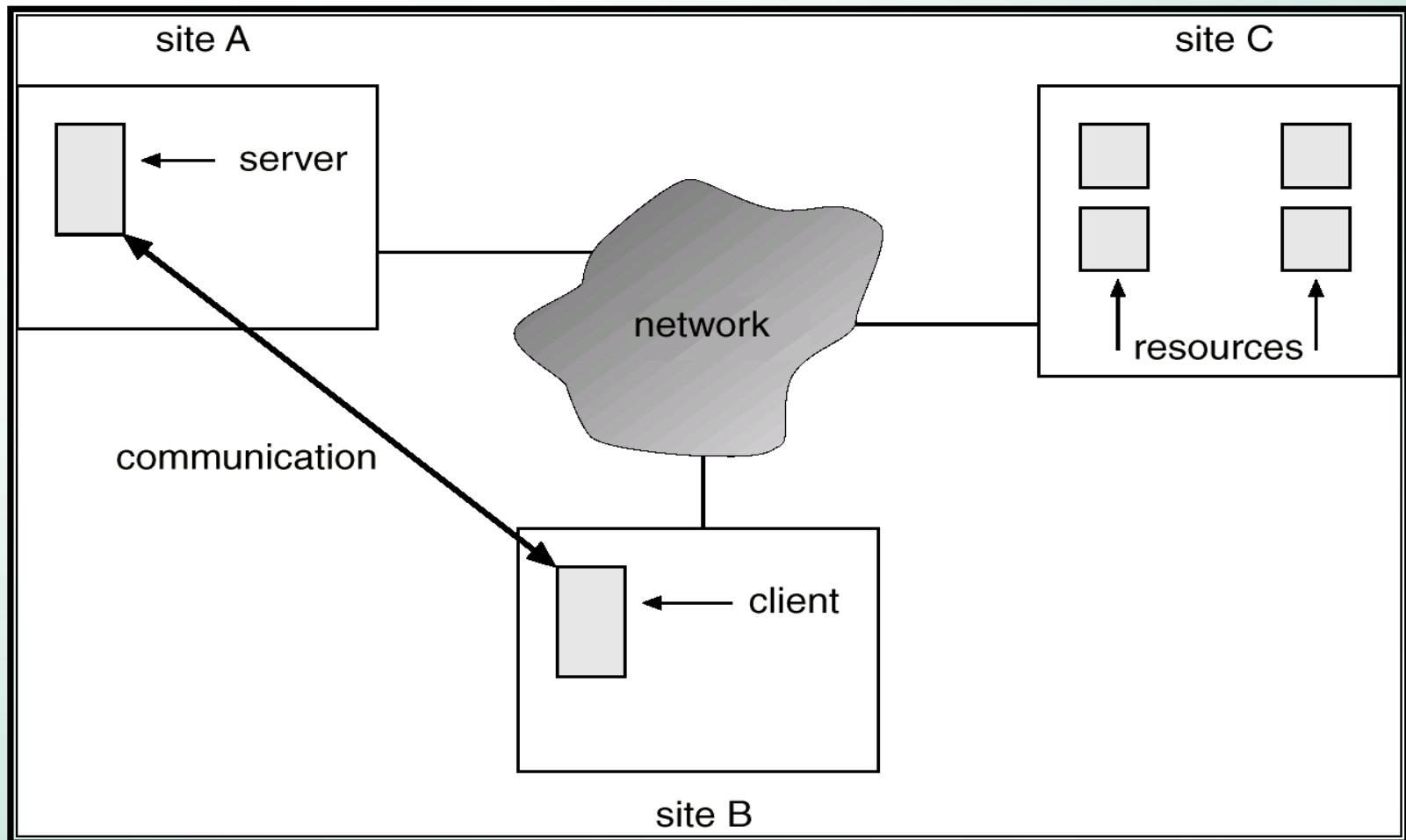
# Πλεονεκτήματα Κατανεμημένων Συστημάτων έναντι ανεξάρτητων PCs

- Διαμοιρασμός δεδομένων και συσκευών
- Αυξημένες δυνατότητες επικοινωνίας ανάμεσα στους ανθρώπους
- Ένα κατανεμημένο σύστημα είναι σαφώς πιο ευέλικτο, αφού μπορούμε να επιτρέπουμε στις διεργασίες των χρηστών να εκτελούνται στον καταλληλότερο υπολογιστή (καλύτερη κατανομή του φόρτου) και η απώλεια ενός μικρού αριθμών μηχανών αντισταθμίζεται από τη δυνατότητα εκτέλεσης των διεργασιών σε οποιαδήποτε άλλη μηχανή

# Μειονεκτήματα Κατανεμημένων Συστημάτων

- Απαιτείται εξειδικευμένο λογισμικό για την υλοποίηση κατανεμημένων συστημάτων
- Σε ποιο βαθμό θα πρέπει να γνωρίζουν οι χρήστες την κατανομή; Πόσα και ποια πράγματα κάνει το σύστημα και ποια οι χρήστες;
- Επικοινωνιακό δίκτυο (απώλειες, κόστος)
- Ασφάλεια

# Ένα Κατανεμημένο Σύστημα





# Πραγματικά Κατανεμημένα Συστήματα

- Στην πραγματικότητα δεν υπάρχει σύστημα που να ικανοποιεί τα κριτήρια για να θεωρηθεί κατανεμημένο σύστημα, παρότι υπάρχουν πολλά υποψήφια
- Η λύση είναι η χρήση ενδιάμεσου λογισμικού
- Αποτελεί ένα πρόσθετο επίπεδο λογισμικού που κρύβει την ετερογένεια σε ένα σύστημα.
- Τοποθετείται ανάμεσα στις εφαρμογές και το δικτυακό λειτουργικό σύστημα

# Μοντέλα Ενδιάμεσου Λογισμικού

- Απλό μοντέλο: όλοι οι πόροι αντιμετωπίζονται ως αρχεία (π.χ. Plan 9)
- Κατανεμημένα συστήματα αρχείων (μόνο για αρχεία αποθήκευσης δεδομένων)
- Κλήση απομακρυσμένων διαδικασιών
- Κατανεμημένων αντικειμένων
- Κατανεμημένων εγγράφων (π.χ. WWW)

# Υπηρεσίες Ενδιάμεσου Λογισμικού

- Υπηρεσίες διαφανούς προσπάθειας
- Υπηρεσίες ονοματολογίας
- Μηχανισμοί διατήρησης για αποθήκευση δεδομένων σε κατακεμημένα συστήματα αρχείων ή σε βάσεις δεδομένων (ενσωματωμένες ή συνδεδεμένες με εφαρμογές μέσω ειδικών μηχανισμών)
- Υπηρεσίες κατακεμημένων συναλλαγών
- Υπηρεσίες ασφάλειας

# Ανοικτή Λειτουργία και Ενδιάμεσο Λογισμικό

- Υπάρχει ανεξαρτησία από το λειτουργικό σύστημα
- Υπάρχει όμως ισχυρή εξάρτηση από το ενδιάμεσο λογισμικό
- Για να υπάρχει διαλειτουργικότητα πρέπει τα interfaces του κατανεμημένου συστήματος να πλήρως ορισμένων προδιαγραφών
- Μη πληρότητα των interfaces οδηγεί σε εφαρμογές που δεν μπορούν να συνεργαστούν

# Σύγκριση μεταξύ συστημάτων

Στοιχείο	Κατανεμημένο ΛΣ		Δικτυακό ΛΣ	ΚΣ με ενδιάμεσο λογισμικό
	Πολυεπεξερ.	Πολυυπολογ.		
Βαθμός διαφάνειας	Πολύ υψηλός	Υψηλός	Χαμηλός	Υψηλός
Ίδιο ΛΣ	Ναι	Ναι	Όχι	Όχι
Πλήθος αντιγράφων ΛΣ	1	N	N	N
Βάση επικοινωνίας	Κοινή μνήμη	Μηνύματα	Αρχεία	Εξαρτάται από το μοντέλο
Διαχείριση πόρων	Καθολική, συγκεντρωτική	Καθολική, κατανεμημένα	Ανά κόμβο	Ανά κόμβο
Επεκτασιμότητα	Όχι	Μέτρια	Ναι	Ποικίλλει
Ανοικτή λειτουργία	Κλειστό	Κλειστό	Ανοικτό	Ανοικτό

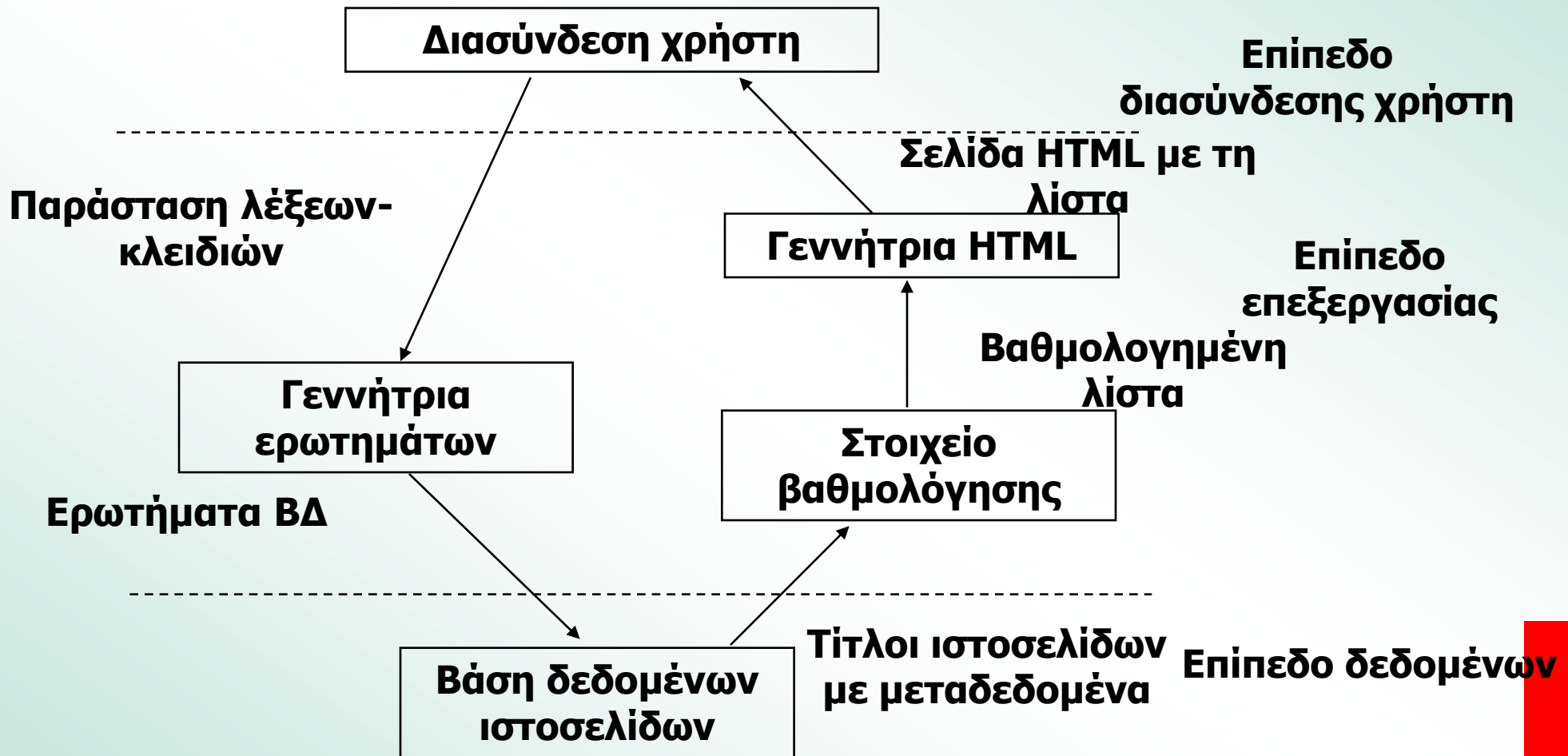
# Το μοντέλο Πελάτη - Εξυπηρετητή

- Εξυπηρετητής: μια διεργασία που υλοποιεί μια υπηρεσία
- Πελάτης: μια διεργασία που ζητά μια υπηρεσία
- Βασίζεται στο πρωτόκολλο ερώτησης – απάντησης
- Το υποκείμενο δίκτυο μπορεί να είναι
  - Connectionless
  - Connection-oriented

# Επίπεδα μιας εφαρμογής

- Επίπεδο διασύνδεσης χρήστη
- Επίπεδο επεξεργασίας
- Επίπεδο δεδομένων

# Επίπεδα μιας εφαρμογής (συνέχεια)





# Απλή αρχιτεκτονική πελάτη - εξυπηρετητή

- Μια μηχανή πελάτη που περιέχει τα προγράμματα που υλοποιούν το επίπεδο διασύνδεσης
- Μια μηχανή εξυπηρετητή που περιέχει τα υπόλοιπα
- Ουσιαστικά ο πελάτης είναι «κουτός»
- Δεν είναι πραγματικά κατανεμημένο μοντέλο

# Τριστρωματική (3-tier) αρχιτεκτονική πελάτη - εξυπηρετητή

- Τα προγράμματα που αποτελούν μέρος του επιπέδου επεξεργασίας βρίσκονται σε ξεχωριστό εξυπηρετητή
- Υπάρχει δυνατότητα να είναι κατά ένα μέρος κατανεμημένα μεταξύ πελάτη και εξυπηρετητή
- Παράδειγμα: η επεξεργασία συναλλαγών
  - Μια ξεχωριστή διεργασία ελεγκτής συναλλαγών συντονίζει όλες τις συναλλαγές σε διαφορετικούς εξυπηρετητές δεδομένων

# Σύγχρονες αρχιτεκτονικές

- Κατακόρυφη κατανομή: κάθε λογικά διαφορετικό στοιχείο τοποθετείται σε άλλη μηχανή (π.χ. τριστρωματική αρχιτεκτονική)
- Οριζόντια κατανομή: ένας πελάτης ή εξυπηρετητής μπορεί να υποδιαιρείται από φυσική άποψη σε λογικά ισοδύναμα μέρη και το κάθε μέρος εργάζεται στο δικό του κομμάτι δεδομένων, μοιράζοντας έτσι το φόρτο (π.χ. Mirror WWW servers)
- Ομότιμη (peer-to-peer) κατανομή