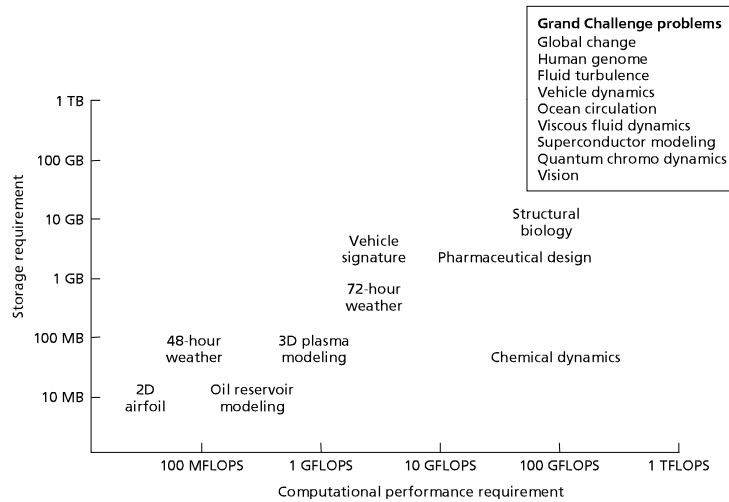


Εισαγωγή

Ανάγκη για Υπολογιστική δύναμη

- Οι ανάγκη για όλο και μεγαλύτερη ταχύτητα στους υπολογισμούς δε μπορεί να ικανοποιηθεί πλέον από μόνο-επεξεργαστικά συστήματα.
- Οι περιοχές που απαιτούν υψηλότερες ταχύτητες υπολογισμού είναι μεταξύ άλλων οι περιοχές του επιστημονικού υπολογισμού, η προσομοίωση επιστημονικών προβλημάτων καθώς και προβλημάτων μηχανικής.
- Σε τέτοια προβλήματα, οι υπολογισμοί πρέπει να ολοκληρώνονται σε ένα εύλογο χρονικό διάστημα.

Προβλήματα με μεγάλο όγκο επεξεργασίας



Τα προβλήματα αυτά είναι επίσης γνωστά και ως «Grand Challenge problems»

Πρόβλεψη Καιρού

- Για την μοντελοποίηση της ατμόσφαιρας, ο τρισδιάστατος χώρος χωρίζεται σε μικρές τρισδιάστατες κυψέλες.
- Πολύπλοκες μαθηματικές εξισώσεις χρησιμοποιούνται για να περιγράψουν τα διάφορα φαινόμενα της ατμόσφαιρας
- Ουσιαστικά, οι συνθήκες σε κάθε κυψέλη (θερμοκρασία, πίεση, υγρασία, ταχύτητα αέρα και κατεύθυνση) υπολογίζονται σε τακτά χρονικά διαστήματα χρησιμοποιώντας τις μετρήσεις της κυψέλης και των κοντινών κυψελών από τον τελευταίο υπολογισμό
- Ο υπολογισμός σε κάθε κυψέλη επαναλαμβάνεται πολλές φορές προκειμένου να προσομοιωθεί ο χρόνος

Παράδειγμα πρόβλεψης καιρού

- Υποθέτουμε ότι όλη η ατμόσφαιρα της γης χωρίζεται σε κυψέλες διαστάσεων $1 \text{ Km} \times 1 \text{ Km} \times 1 \text{ Km}$ μέχρι το ύψος των 10 Km (10 κυψέλες σε ύψος).
- Συνολικά, θα έχουμε 5×10^8 κυψέλες.
- Υποθέτουμε κάθε υπολογισμός απαιτεί 200 λειτουργίες κινητής υποδιαστολής. Σε ένα βήμα υπολογισμού, 10^{11} τέτοιες λειτουργίες είναι απαραίτητες.
- Για να προβλέψουμε τον καιρό για μετά από 7 ημέρες χρησιμοποιώντας διαστήματα του ενός λεπτού, ένας υπολογιστής που εκτελεί 1 Gflops (10^9 λειτουργίες κινητής υποδιαστολής/δευτερόλεπτο) χρειάζεται 10^6 δευτερόλεπτα ή πάνω από 10 ημέρες.
- Για να εκτελέσουμε ένα υπολογισμό σε 5 λεπτά απαιτούνται υπολογιστές που λειτουργούν στα 3.4 Tflops (3.4×10^{12} λειτουργίες κινητής υποδιαστολής/δευτερόλεπτο).

Μοντελοποίηση της κίνησης Ουρανίων Σωμάτων

- Κάθε σώμα έλκεται από κάθε άλλο σώμα από τους νόμους του Νεύτωνα.
- Η ελκτική δύναμη μεταξύ δύο σωμάτων είναι αντιστρόφως ανάλογη του τετραγώνου της απόστασής τους.
- Η κίνηση κάθε σώματος καθορίζεται από τη συνισταμένη των δυνάμεων που ασκούνται σε κάθε σώμα.
- Αν έχουμε N σώματα, πρέπει να υπολογίσουμε $N - 1$ δυνάμεις για κάθε σώμα και συνολικά σχεδόν N^2 calculations.
- Χρησιμοποιώντας ένα καλό προσεγγιστικό αλγόριθμο για το συγκεκριμένο πρόβλημα μπορούμε να χαμηλώσουμε τη πολυπλοκότητα σε $N \log_2 N$
- Μετά τη μετακίνηση των σωμάτων στη νέα τους θέση τους, ο παραπάνω υπολογισμός πρέπει να επαναληφθεί αφού οι ασκούμενες δυνάμεις μεταξύ των σωμάτων έχουν αλλάξει.
- Για παράδειγμα: ένας γαλαξίας μπορεί να έχει 10^{11} αστέρια.
- Ακόμα και αν υποθέσουμε ότι κάθε υπολογισμός χρειάζεται ένα 1 ms για να ολοκληρωθεί, συνολικά χρειάζονται 10^9 χρόνια για κάθε επανάληψη χρησιμοποιώντας τον αλγόριθμο πολυπλοκότητας $O(N^2)$ ή σχεδόν ένα χρόνο για μία επανάληψη χρησιμοποιώντας τον αλγόριθμο πολυπλοκότητας $O(N \log_2 N)$.

Παράλληλος Υπολογισμός

- Επίλυση υπολογιστικών προβλημάτων με τη χρήση περισσότερων από ένα υπολογιστή ή ενός υπολογιστής με περισσότερους από ένα επεξεργαστή
- Το βασικό κίνητρο είναι συνήθως η ολοκλήρωση των υπολογισμών σε συντομότερο χρόνο.
- Ιδανικά, με n υπολογιστικά στοιχεία που δουλεύουν ταυτόχρονα, ο συνολικός χρόνος υπολογισμού μειώνεται n φορές.
- Στη πράξη, η επιτάχυνση στον υπολογισμό θα είναι μικρότερη από n .
- Άλλα κίνητρα για παράλληλο υπολογισμό είναι η ανοχή στα σφάλματα, περισσότερη διαθέσιμη μνήμη κτλ.

Παράγοντας επιτάχυνσης

- Ο παράγοντας επιτάχυνσης ορίζεται ως

$$S(p) = \frac{\text{Χρόνος εκτέλεσης χρησιμοποιώντας ένα επεξεργαστή (ο καλύτερος ακολουθιακός αλγόριθμος)}}{\text{Χρόνος εκτέλεσης χρησιμοποιώντας ένα πολυεπεξεργαστή με } p \text{ επεξεργαστές}} = \frac{t_s}{t_p}$$

όπου t_s είναι ο χρόνος εκτέλεσης σε ένα επεξεργαστή και t_p είναι ο χρόνος εκτέλεσης σε ένα πολυεπεξεργαστή

- Ο παράγοντας επιτάχυνσης $S(p)$ δίνει την αύξηση στη ταχύτητα υπολογισμού με τη χρήση επεξεργαστή
- Για το μονό επεξεργαστή, χρησιμοποιούμε το καλύτερο ακολουθιακό αλγόριθμο.
- Ο αλγόριθμος στη παράλληλη υλοποίηση μπορεί να είναι και συνήθως είναι διαφορετικός

Μέγιστη επιτάχυνση

- Η μέγιστη επιτάχυνση που μπορούμε να επιτύχουμε με p επεξεργαστές είναι συνήθως p (γραμμική επιτάχυνση).
- Είναι επίσης πιθανόν να λάβουμε υπερ-γραμμική (superlinear) επιτάχυνση (μεγαλύτερη από p). Αυτό συνήθως οφείλεται:
- Στην επιπλέον μνήμη του πολυεπεξεργαστικού συστήματος multiprocessor system
- ή/και στην καλύτερη απόδοση των cache μνημών στα πολυεπεξεργαστικά συστήματα

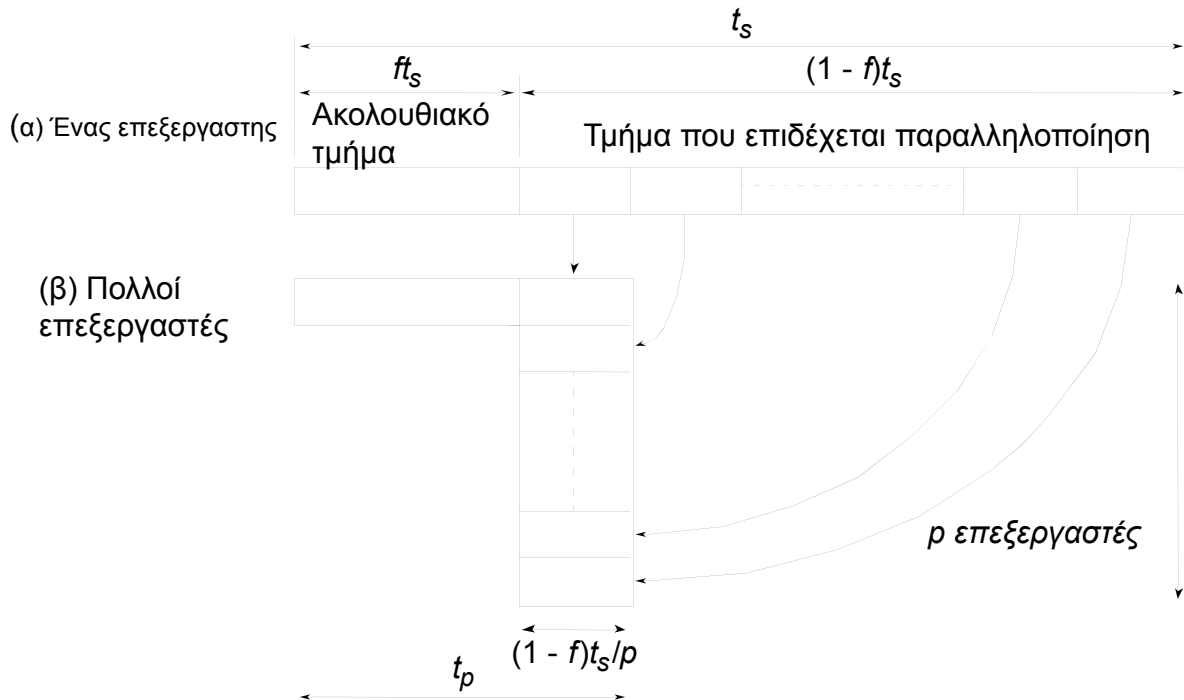
Μέγιστη επιτάχυνση Νόμος του Amdahl

- Σχεδόν σε όλους τους υπολογισμούς υπάρχει ένα τμήμα που είναι ακολουθιακό και ένα τμήμα το οποίο μπορεί να διαιρεθεί σε επιμέρους υποέργα.
- Έστω t_s ο συνολικός χρόνος της ακολουθιακής εκτέλεσης του υπολογισμού και έστω ft_s ο χρόνος εκτέλεσης του πρώτου τμήματος και $(1-f)t_s$ ο χρόνος εκτέλεσης του δεύτερου τμήματος
- Σε ένα πολυεπεξεργαστή με p επεξεργαστές, το πρώτο τμήμα εξακολουθεί να έχει χρόνο εκτέλεσης ft_s ενώ το δεύτερο τμήμα εκτελείται σε χρόνο $(1-f)t_s/p$
- Άρα η επιτάχυνση για αυτό τον υπολογισμό δίνεται από την ακόλουθη σχέση:

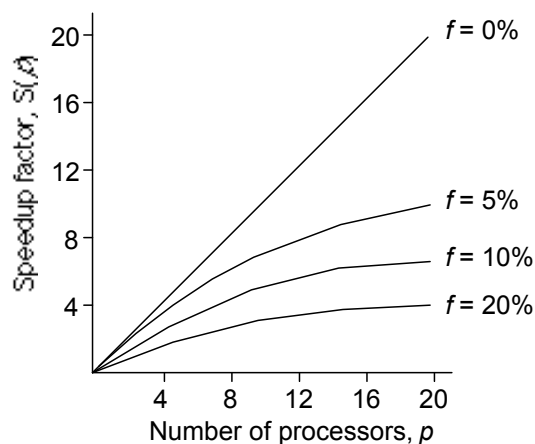
$$S(p) = \frac{t_s}{ft_s + (1-f)t_s/p} = \frac{p}{1 + (p-1)f}$$

- Ο παραπάνω τύπος είναι γνωστός ως νόμος του Amdahl

Μέγιστη επιτάχυνση Νόμος του Amdahl



Επιτάχυνση για διαφορετικό πλήθος επεξεργαστών



- Ακόμα και με άπειρο πλήθος επεξεργαστών η μέγιστη επιτάχυνση είναι το πολύ $1/f$
- Παράδειγμα: Με $f=5\%$, η μέγιστη επιτάχυνση είναι 20, ανεξάρτητα του πλήθους των επεξεργαστών

Παράδειγμα Υπεργραμμικής Επιτάχυνσης – Αλγόριθμοι Αναζήτησης

- Σε ένα απλό αλγόριθμο που επιλύει ένα πρόβλημα αναζήτησης, ο χώρος των λύσεων εξετάζεται εξαντλητικά.
- Υποθέτουμε ο χώρος των λύσεων χωρίζεται σε p υποχώρους και αυτοί μοιράζονται στους p επεξεργαστές και κάθε ένας επεξεργαστής εκτελεί μία ανεξάρτητη αναζήτηση στον υποχώρο που του αντιστοιχεί.
- Έστω ότι η λύση βρίσκεται στο $x+1$ υποχώρο και μάλιστα μπορεί να βρεθεί σε Δt βήματα ακολουθιακά σε αυτό τον υποχώρο.
- Η συνολική επιτάχυνση θα είναι:

$$S(p) = (x t_s / p + \Delta t) / \Delta t$$

- Η χειρότερη περίπτωση για τον ακολουθιακό αλγόριθμο συμβαίνει όταν η λύση βρίσκεται στο τελευταίο υποχώρο:

$$S(p) = ((p-1)t_s / p + \Delta t) / \Delta t \rightarrow \infty$$

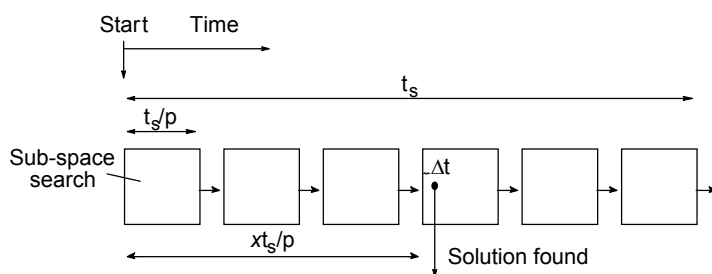
καθώς το $\Delta t \rightarrow 0$

- Ελάχιστη επιτάχυνση θα έχουμε όταν η λύση βρίσκεται στο πρώτο υποχώρο:

$$S(p) = \Delta t / \Delta t = 1$$

- Στη πράξη, η επιτάχυνση εξαρτάται από τον υποχώρο που περιέχει τη λύση

Παράδειγμα Υπεργραμμικής Επιτάχυνσης – Αλγόριθμοι Αναζήτησης

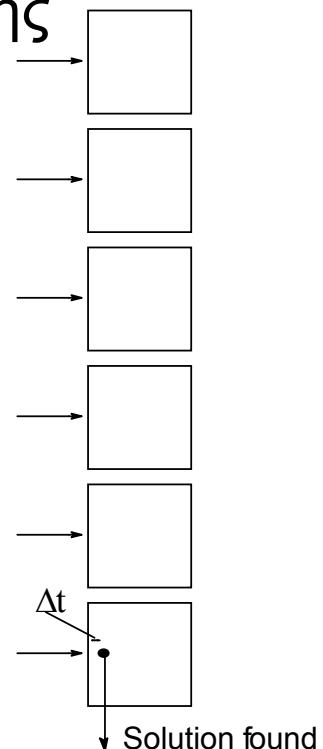


x indeterminate

Ακολουθιακή εκτέλεση
Συνολικός χρόνος: $x t_s / p + \Delta t$

Επιτάχυνση $S(p) = (x t_s / p + \Delta t) / \Delta t$

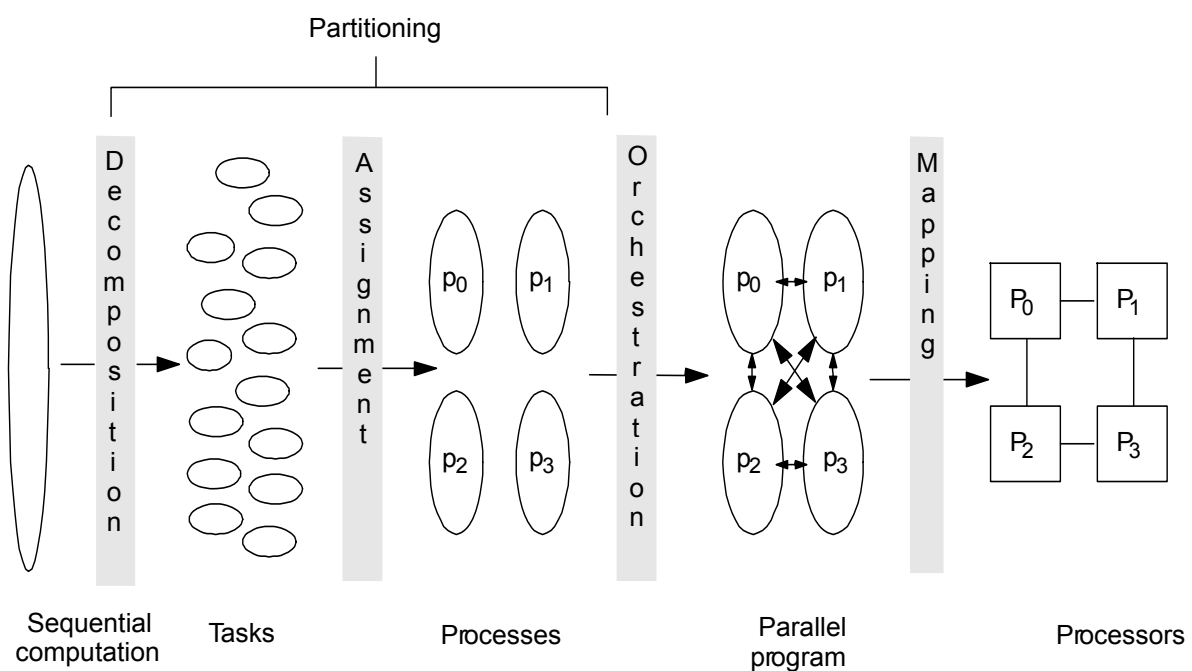
Παράλληλη
εκτέλεση:
 $T_p(p) = \Delta t$



Παράλληλος Προγραμματισμός

- Παράλληλο Πρόγραμμα: περιέχει δύο ή περισσότερες διεργασίες (processes), οι οποίες εργάζονται μαζί για την επίλυση του ίδιου προβλήματος.
- Διεργασία (Process): Ακολουθιακό πρόγραμμα (ακολουθία εντολών)
- Οι πολλαπλές διεργασίες ενός παράλληλου προγράμματος συνεργάζονται επικοινωνώντας η μια με την άλλη.
- Η επικοινωνία πραγματοποιείται με τη χρήση
 - Κοινών ή διαμοιραζόμενων μεταβλητών (shared variables) ή
 - Μέσω αποστολής και παραλαβής μηνυμάτων (message passing)

Στάδια Δημιουργίας Παράλληλου Προγράμματος

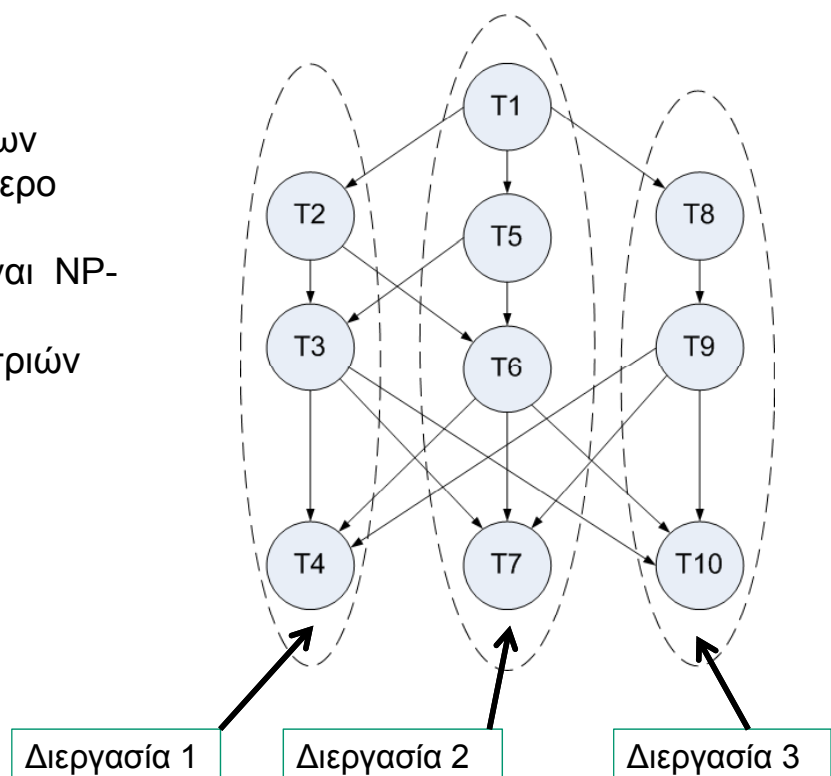


Στάδια Δημιουργίας Παράλληλου Προγράμματος

- Ο ακολουθιακός υπολογισμός διαιρείται σε υποέργα
- Τα υποέργα μοιράζονται σε πλήθος διεργασιών
- Στις περισσότερες περιπτώσεις, ένα υποέργο χρειάζεται τα αποτελέσματα της επεξεργασίας ενός άλλου υποέργου
- Αυτό απαιτεί συντονισμό μεταξύ των διεργασιών του παράλληλου προγράμματος
- Η συνεργασία μεταξύ των διεργασιών απαιτεί επικοινωνία μεταξύ τους.
- Το τελευταίο βήμα είναι η αντιστοίχιση των διεργασιών στους επεξεργαστές ενός πολυεπεξεργαστικού συστήματος
- Κάθε επεξεργαστής εκτελεί τις διεργασίες που του έχουν ανατεθεί.

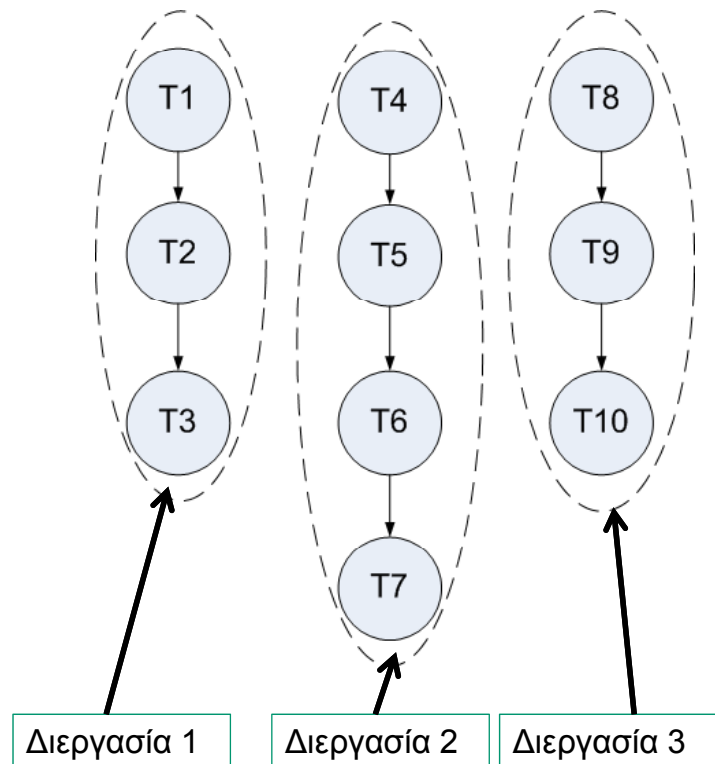
Παράλληλη εκτέλεση εργασιών

- Τελικός στόχος είναι η ολοκλήρωση των όλων των υποέργων στο συντομότερο δυνατόν χρόνο
- Γενικά, το πρόβλημα είναι NP-complete
- Κάθε ακμή μεταξύ των τριών διαμερίσεων δηλώνει επικοινωνία μεταξύ των αντίστοιχων διεργασιών



Ένα παράδειγμα εύκολου παραλληλισμού

- Σε ορισμένα προβλήματα δεν υπάρχουν εξαρτήσεις μεταξύ των υποέργων
- Σε αυτή τη περίπτωση, έχουμε ιδανική επιτάχυνση (Embarrassingly parallel program)



Επικοινωνία στις Παράλληλες Μηχανές

- Υπάρχουν δύο τρόποι επικοινωνίας των επεξεργαστών ενός παράλληλου συστήματος:
 1. Η χρήση κοινής ή διαμοιραζόμενης μνήμης (*shared memory*).
 2. Η επικοινωνία των επεξεργαστών μέσω ανταλλαγής ή περάσματος μηνυμάτων (*message passing*).
- Αντίστοιχα έχουμε:
 - Συστήματα τα οποία διαθέτουν κοινό χώρο διευθύνσεων (*shared dataspace*): Πολυεπεξεργαστές – Συστήματα Κοινής Μνήμης (*Shared-memory systems - multiprocessors*)
- Συστήματα τα οποία υποστηρίζουν πέρασμα μηνυμάτων. Έχουμε δύο περιπτώσεις:
 - Πολυυπολογιστές - Συστήματα Κατανεμημένης Μνήμης (*Distributed-memory systems - multicomputers*)
 - Clusters (*Networks of Workstations (NoWs) and Beowulf Clusters*)

Πολυεπεξεργαστές (Multiprocessors)

- Αποτελούνται από έναν αριθμό επεξεργαστών οι οποίοι έχουν πρόσβαση στην ίδια κοινή μνήμη.

Υπάρχουν δύο κατηγορίες πολυεπεξεργαστών:

- Πολυεπεξεργαστές *Ομοιόμορφης Προσπέλασης Μνήμης (Uniform Memory Access – UMA)* στους οποίους η κοινή μνήμη είναι κεντρική.
- Πολυεπεξεργαστές *Μη Ομοιόμορφης Προσπέλασης Μνήμης (Non-Uniform Memory Access – NUMA)* στους οποίους η κοινή μνήμη είναι κατανομημένη στους επεξεργαστές.

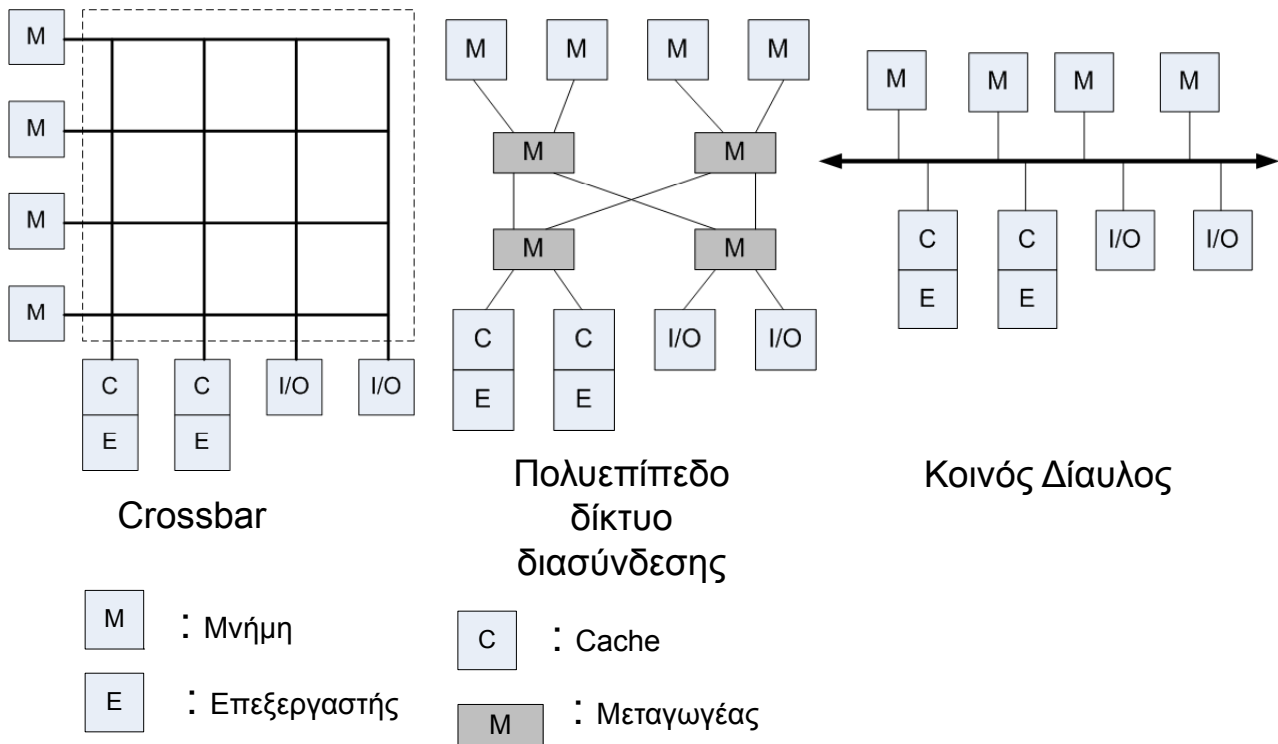
Πολυεπεξεργαστές Κοινής Μνήμης

Οι επεξεργαστές έχουν πρόσβαση σε μία κεντρική κοινή μνήμη μέσω ενός κεντρικού μηχανισμού προσπέλασης μνήμης.



- Η επικοινωνία των επεξεργαστών γίνεται μέσω των συμβατικών εντολών πρόσβασης μνήμης (load & store)
- Μέρος της εικονικής μνήμης των διεργασιών (processes) είναι κοινό δηλ. αντιστοιχεί σε κοινό χώρο της μνήμης

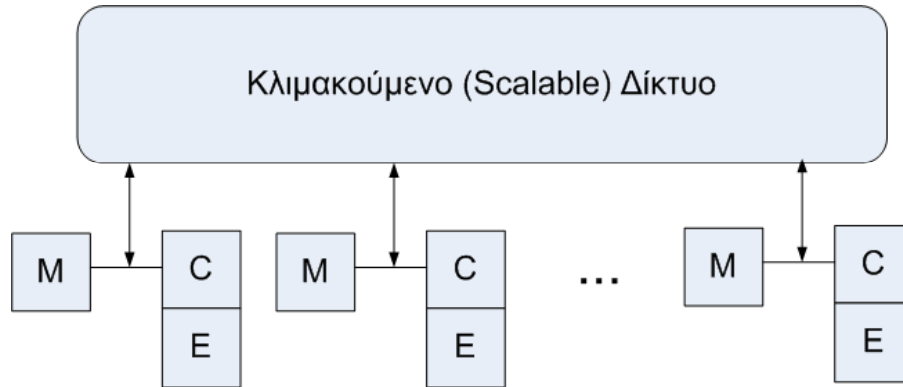
Τυπικά σχήματα διασύνδεσης μνημών – επεξεργαστών



Τυπικά σχήματα διασύνδεσης μνημών – επεξεργαστών

- Το δίκτυο crossbar παρέχει ξεχωριστή ζεύξη για κάθε ζεύγος επεξεργαστή μνήμη. Είναι λύση με υψηλό κόστος
- Ο δίαυλος είναι η πιο οικονομική λύση αλλά δημιουργεί συχνά φαινόμενα συμφόρησης αφού όλες οι επικοινωνίες πραγματοποιούνται μέσω του κοινού διαύλου.
- Μία ενδιάμεση λύση από πλευράς κόστους και απόδοσης είναι το πολυεπίπεδο δίκτυο διασύνδεσης.

Πολυεπεξεργαστές Μη Ομοιόμορφης Προσπέλασης Μνήμης (NUMA)



M: Μνήμη
C: Cache
E: Επεξεργαστής

- Κάθε επεξεργαστής διαθέτει τη δική του μνήμη
- Δίνεται η ψευδαίσθηση ότι υπάρχει κοινή διαμοιραζόμενη μνήμη.
- Όμως ο χρόνος για την ολοκλήρωση μιας εγγραφής ή ανάγνωσης διαφέρει σημαντικά ανάλογα αν η λειτουργία στη τοπική μνήμη ή σε μνήμη άλλου επεξεργαστή

Προγραμματισμός Πολυεπεξεργαστών (Συστήματα Διαμοιραζόμενης Μνήμης)

Ο προγραμματισμός σε τέτοια συστήματα γίνεται με τους ακόλουθους τρόπους:

- Νήματα (Threads) – ο προγραμματιστής χωρίζει το πρόγραμμα σε ξεχωριστές παράλληλες ακολουθίες εντολών (threads). Κάθε νήμα μπορεί να προσπελάζει σφαιρικές μεταβλητές δηλωμένες έξω από κάθε νήμα.

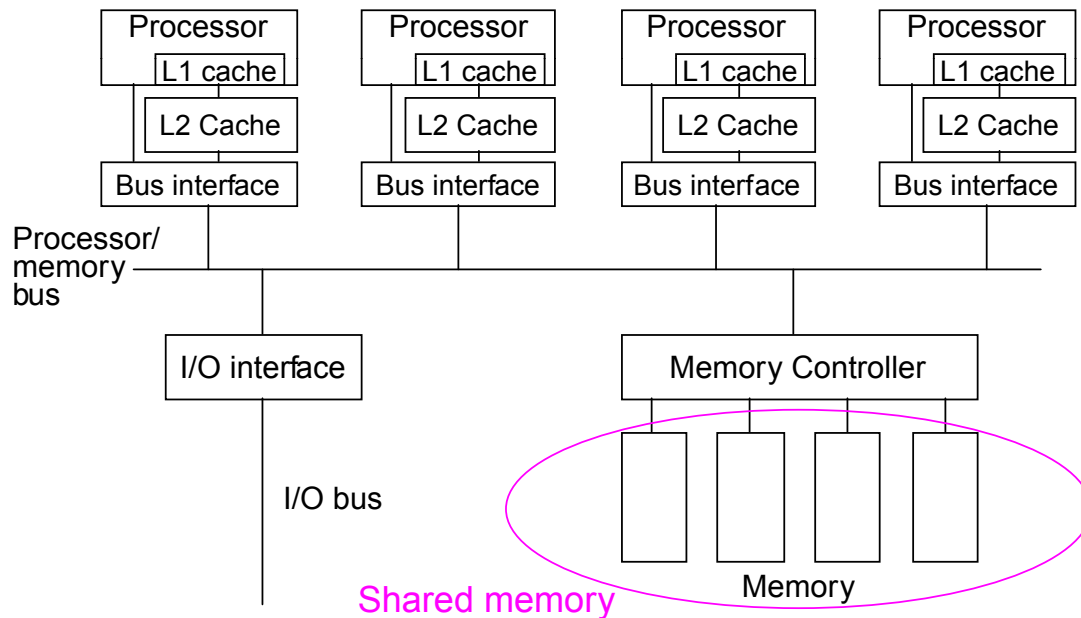
Παράδειγματα: Pthreads, Java threads, CUDA (γλώσσα προγραμματισμού των καρτών NVIDIA)

- Σε μία «κλασσική» γλώσσα προγραμματισμού εισάγονται οδηγίες προεπεξεργασίας (directives) προς το μεταγλωττιστή οι οποίες δηλώνουν κοινές μεταβλητές και προσδιορίζουν παραλληλισμό.

Παράδειγμα: η OpenMP η οποία αποτελεί πρότυπο για το προγραμματισμό πολυεπεξεργαστών

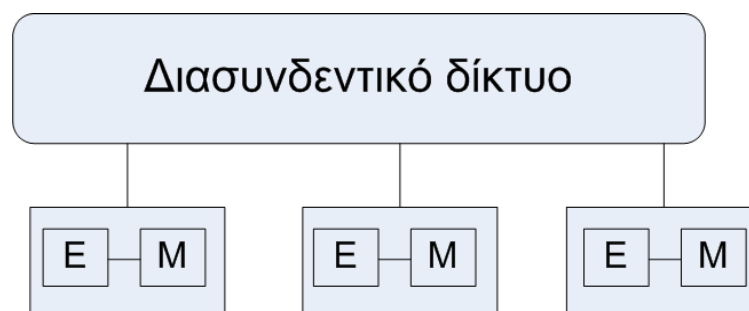
- Ο προγραμματιστής γράφει σε μία «κλασσική» γλώσσα προγραμματισμού και στη συνέχεια ο μεταγλωττιστής μετατρέπει το πρόγραμμα σε παράλληλο. Γενικά, η αυτόματη παραλληλοποίηση είναι δύσκολο ζήτημα και δεν επιτυγχάνει πάντα αποδοτικές λύσεις

Quad Pentium Shared Memory Multiprocessor



Πολυυπολογιστές (Multicomputers) Συστήματα Κατανεμημένης Μνήμης

- Κάθε επεξεργαστής έχει τη δική του τοπική μνήμη. Δεν υπάρχει κοινή μνήμη στην οποία έχουν πρόσβαση όλοι οι επεξεργαστές.
- Η επικοινωνία των επεξεργαστών γίνεται με πέρασμα μηνυμάτων (message-passing) μέσω δικτύου διασύνδεσης.



E:
Επεξεργαστής
M: Μνήμη