ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

ΜΕΤΡΑ ΑΠΟΔΟΣΗΣ & ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΥΨΗΛΩΝ ΕΠΙΔΟΣΕΩΝ

CEID
Parallel Processing
Programming & Architecture

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Η υλοποίηση ενός παράλληλου αλγόριθμού σε πολυεπεξεργαστικό σύστημα ***συγκεκριμένης αρχιτεκτονικής***, απαιτεί κατάλληλο ταίριασμα των υπολογισμών με τους διαθέσιμους πόρους του πολυεπεξεργαστικού συστήματος.

Στόχος αυτής της προσπάθειας, του ταιριάσματος δηλαδή των χαρακτηριστικών του παράλληλου αλγορίθμου με τα χαρακτηριστικά της μηχανής, είναι η βελτιστοποίηση διαφόρων μέτρων επίδοσης, όπως:

- **Χρονοβελτίωση**
- **Αποδοτικότητα**
- **κόστος**

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Για την επίτευξη του παραπάνω στόχων πρέπει πρώτα να εντοπιστούν οι υπάρχουσες εξαρτήσεις μεταξύ των «πράξεων-υπολογισμών» και οι δυνατότητες παράλληλης εκτέλεσης τμημάτων του αλγορίθμου.

Ο Βαθμός Παραλληλισμού

Βαθμός παραλληλισμού (β.π.) ΒΠ(t), είναι ο αριθμός των ωφέλιμων πράξεων, υπολογισμοί, που μπορούν να εκτελεστούν ταυτόχρονα κατά τη χρονική στιγμή (t), από πολυεπεξεραστική μηχανή συγκεκριμένης αρχιτεκτονικής.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Ο βαθμός παραλληλισμού *ΒΠ(t)* όμως είναι χαρακτηριστικό του αλγορίθμου.

Συνήθως οι χρονικές στιγμές ενδιαφέροντος (βήματα εκτέλεσης) ενός παράλληλου αλγορίθμου είναι διακριτές $t_0,\ldots, t_k$, έτσι θα συμβολίζουμε το β.π. με, $ΒΠ(t_k)=B_k$.

Αν δηλαδή κατά τη χρονική στιγμή *t* υπάρχουν *ΒΠ(t)* επεξεργαστές διαθέσιμοι, τότε όλες οι *ΒΠ(t)* πράξεις που ανήκουν στον ίδιο αλγόριθμο μπορούν να εκτελεστούν στο χρόνο που απαιτείται για μία μόνο πράξη.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Βέβαια θεωρούμε εδώ ότι όλες οι *ΒΠ(t)* πράξεις είναι των ιδίων υπολογιστικών απαιτήσεων.

Έτσι μεγάλος *ΒΠ*, για τον ίδιο αλγόριθμο, σημαίνει και μεγάλη μείωση του χρόνου εκτέλεσης των πράξεων του αλγορίθμου, αν φυσικά υπάρχουν διαθέσιμοι και οι απαιτούμενοι υπολογιστικοί πόροι. Ενώ μικρός *ΒΠ*, για τον ίδιο αλγόριθμο, σημαίνει αδυναμία εκτέλεσης πολλών πράξεων παράλληλα, αν υπάρχου διαθέσιμοι πόροι.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ

ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Εδώ αντιμετωπίζουμε ένα πολύ σημαντικό πρόβλημα:

- Επειδή ο *ΒΠ* σε ένα πραγματικό πρόβλημα μεταβάλλεται συνεχώς κατά τα διάφορα βήματα εκτέλεσης του αλγορίθμου (στιγμιότυπα). Σε ένα αλγόριθμο με μεγάλο *ΒΠ,* θα υπάρξουν κατά τη διάρκεια της εκτέλεσής του, στιγμές που θα χρησιμοποιεί όλους τους διαθέσιμους πόρους του συστήματος (*P=ΒΠ(t), P επεξεργαστές)* και στιγμιότυπα του, με πολλούς ανενεργούς επεξεργαστές *(P- ΒΠ(t’)*.

Συνεπώς ο μεγάλος *ΒΠ* μπορεί να προκαλέσει μικρή **αξιοποίηση** του συστήματος, και κατά συνέπεια μικρή **αποδοτικότητα**.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Δηλαδή, η επιθυμία για μεγάλη μείωση του χρόνου εκτέλεσης ενός αλγορίθμου μπορεί να μας οδηγήσει στην υλοποίηση αλγορίθμου με μεγάλο *ΒΠ* για συγκεκριμένα χρονικά διαστήματα, με συνέπεια τη χρήση πολύ μεγάλου αριθμού επεξεργαστών ⇒ ανενεργοί πόροι του συστήματος για μεγάλα χρονικά διαστήματα ⇒ μικρή αποδοτικότητα με μεγάλο *ΒΠ*.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Ένα επιπλέον πρόβλημα, είναι ότι η χρήση μεγάλου αριθμού επεξεργαστών ακόμα και όταν ο *ΒΠ* της εφαρμογής παραμένει μεγάλος για το μεγαλύτερο χρονικό διάστημα, θα μας δημιουργήσει πιθανότατα πρόβλημα μείωσης της απόδοσης του συστήματος και κατ΄επέκταση μείωση του χρόνου εκτέλεσης της εφαρμογής, λόγω **επιβαρύνσεων** $P>1$.

Δηλαδή στη σχέση

$$T(n,P)=\Omega(n,P)*\tau_{\alpha}+E(n,P)*\tau_{\varepsilon}$$

ισχύει γενικά ότι η Ω είναι φθίνουσα συνάρτηση του P (P, αριθμός επεξεργαστών που λαμβάνουν μέρος στην εκτέλεση της εφαρμογής). Ταυτόχρονα όμως ισχύει και ότι το Ε είναι αύξουσα συνάρτηση του P.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Καταλήγουμε τελικά στο συμπέρασμα, ότι στην πράξη απαιτείται ιδιαίτερη προσοχή στην επιλογή του P ώστε συνολικά να μειωθεί ο παράλληλος χρόνος εκτέλεσης ενός αλγορίθμου, ταυτόχρονα όμως προσπαθούμε να υλοποιούμε μεγάλο *ΒΠ* άλλα πάντα σε συνάρτηση με τους διαθέσιμους πόρους, αλλά και τις επιβαρύνσεις που θα επιφέρει ο *ΒΠ*.

Ως γενική παρατήρηση τελικά, μπορούμε να πούμε ότι στόχος μας είναι:

ο βέλτιστος χρόνος εκτέλεσης ενός παράλληλου αλγορίθμου με ταυτόχρονή φροντίδα για την όσο το δυνατό καλύτερη χρήση των πόρων του συστήματος.

Τα δύο αυτά μεγέθη είναι συνήθως αντικρουόμενα και έτσι η επίτευξή βέλτιστης λύσης είναι αδύνατη. Οι λύσεις που δίνουμε στο πρόβλημα αυτό είναι προσεγγιστικές με χρήση ευρετικών αλγορίθμων (heuristic algorithms).

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΚΑΤΑΜΕΡΙΣΜΟΣ

•Στην περίπτωση εκτέλεσης παράλληλου αλγορίθμου σε παράλληλο υπολογιστικό σύστημα, ο αριθμός των επεξεργαστών καθορίζει (συνήθως) και τον αριθμό των πράξεων που μοιράζονται στον καθένα. Έτσι για παράδειγμα στην περίπτωση ισοκατανομής των πράξεων, κάθε επεξεργαστής θα εκτελέσει περίπου $\beta(t)/P$ πράξεις. Ο αριθμός αυτός εκπροσωπεί την λεπτότητα του καταμερισμού των υπολογισμών (granularity).

•Όσο πιο μεγάλη κατάτμηση ενός αλγορίθμου έχουμε, μεγάλος *ΒΠ* (πολλές δηλαδή πράξεις), τόσο μεγαλύτερη είναι και η λεπτότητα του καταμερισμού του αλγορίθμου (fine granularity – fine grain) και αντίστροφα, όσο μικρότερη κατάτμηση τόσο πιο αδρό καταμερισμό (coarse granularity – coarse grain) θα έχουμε.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΚΑΤΑΝΟΜΗ ΥΠΟΛΟΓΙΣΜΩΝ

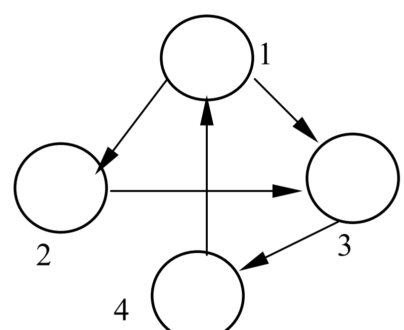
•Για την κατανομή των υπολογισμών ενός αλγορίθμου σε ένα παράλληλο σύστημα, χρησιμοποιούμε συχνά την αποτύπωση των πράξεων του μέσω ενός Διευθυνόμενου Άκυκλου Γραφήματος (ΔΑΓ).

•Γράφημα είναι ένα πεπερασμένο σύνολο κόμβων ή κορυφών V και τόξων ή πλευρών S, όπου ο συμβολισμός $e_{ij} \in S$, εκπροσωπεί μια σχέση μεταξύ των κορυφών v_i, v_j. Στην περίπτωση μας οι κορυφές εκπροσωπούν πράξεις (οι οποίες μπορεί να είναι και σύνθετα έργα, π.χ. βρόχοι-loops), ενώ οι πλευρές εκφράζουν προτεραιότητες μεταξύ των κορυφών, έργων. Όταν η σχέση μεταξύ των v_i, v_j δεν είναι αμφίδρομη, και η πλευρά έχει διεύθυνση από το v_i στο v_j ή αντίστροφα όχι όμως και προς τις δύο κατευθύνσεις τότε το γράφημα καλείται διευθυνόμενο.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΚΑΤΑΝΟΜΗ ΥΠΟΛΟΓΙΣΜΩΝ

Διαδρομή $\Delta(i,j)$ από τον v_i στον v_j είναι μια πεπερασμένη και μη κενή ακολουθία πλευρών $\{e_{ir},...,e_{xj}\}$ όπου για οποιεσδήποτε δύο διαδοχικές πλευρές $\{e_{\alpha\beta}, e_{\gamma\delta}\}$, εάν υπάρχουν, ισχύει $\beta = \gamma$. Έτσι ακολουθώντας τις πλευρές της διαδρομής με την σειρά αυτή, μπορούμε να βαδίσουμε από την v_i στην v_j. Αν $j = i$ τότε η διαδρομή "κλείνει" στη v_i και καλείται κύκλος. Αν δεν υπάρχου κύκλοι έχουνε ένα ΔΑΓ.

Παράδειγμα: Το ακόλουθο γράφημα αντιστοιχεί στο σύνολο των κορυφών $\{v_1, v_2, v_3, v_4\}$ με διευθυνόμενες πλευρές $\{e_{12}, e_{13}, e_{23}, e_{34}, e_{41}\}$. Το γράφημα δεν είναι ΔΑΓ γιατί $\exists\ \{e_{12}, e_{23}, e_{34}, e_{41}\}$.

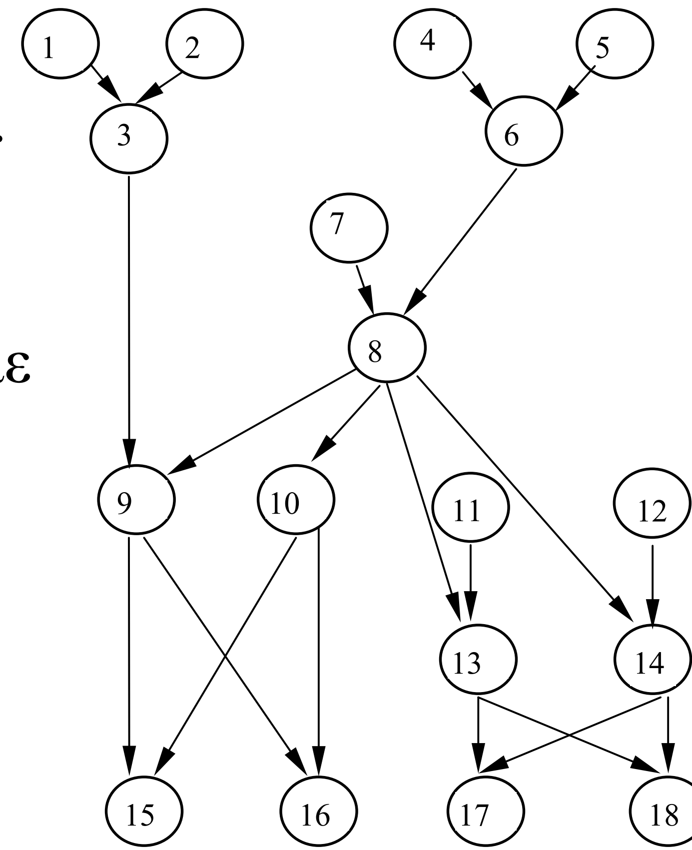


ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ
ΚΑΤΑΝΟΜΗ ΥΠΟΛΟΓΙΣΜΩΝ

Για την καλύτερη κατανόηση των διαφόρων θεμάτων, είναι ίσως χρήσιμη μια "τακτοποίηση" του ΔΑΓ ώστε οι κόμβοι του να ταξινομηθούν σε επίπεδα $k = 0, ..., K$. Τα έργα του επιπέδου k θα μπορούν να αρχίσουν μαζί την χρονική στιγμή t_k. Στο επίπεδο 0 τοποθετούνται έργα που δεν εξαρτώνται από αποτελέσματα άλλων, άρα η εκτέλεσή τους μπορεί ν' αρχίσει αμέσως, την χρονική στιγμή $t_0 = 0$. Επαγωγικά, στο επίπεδο $j >= 1$ τοποθετούνται όλες οι κορυφές / έργα V_r με την ιδιότητα ότι όλες οι πλευρές που φθάνουν στην V_r ξεκινούν από κορυφές των προηγούμενων επιπέδων $k<j$ και τουλάχιστον ένα από αυτά τα επίπεδα είναι το $j-1$.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ

Παράδειγμα: Στο διπλά σχήμα φαίνεται το ΔΑΓ ενός υποθετικού υπολογισμού. Αν εφαρμόσουμε αυτά που προηγουμένως αναφέραμε, με σκοπό να αυξήσουμε το *ΒΠ* και να μειώσουμε το χρόνο εκτέλεσης αυτού του υποθετικού υπολογισμού σε ένα πολυεπεξεργαστικό σύστημα, τότε με κατάλληλη παρέμβαση το δίπλα ΔΑΓ θα τροποποιηθεί όπως φαίνεται στην επόμενη εικόνα.

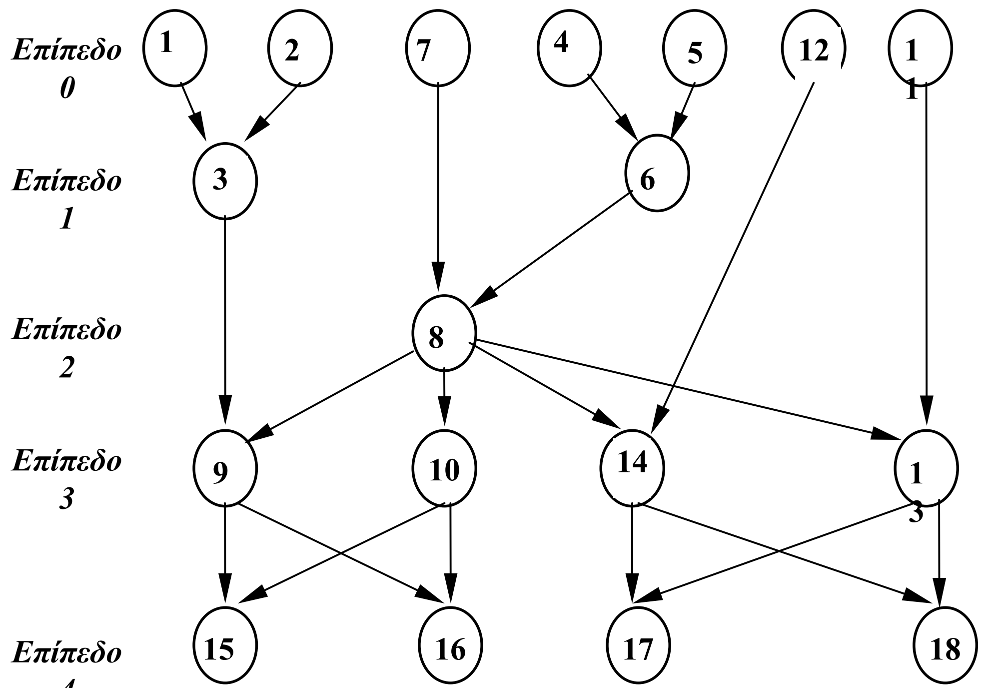


ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ

Ο *ΒΠ* στα διάφορα επίπεδα είναι τώρα 7, 2, 1, 4, 4. Αυτό όμως δεν είναι δεσμευτικό, π.χ. η κορυφή 12.

Σημειώνεται επίσης ότι K είναι το "βάθος" του ΔΑΓ και ότι B_k = *ΒΠ* (t_k) είναι το σύνολο των κόμβων του επιπέδου k. Το σύνολο των πράξεων είναι:

$$\boldsymbol{n}_{//} = \sum_{k=0}^{K} \boldsymbol{B}_k$$



ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ

Επίσης εάν ο μέσος χρόνος τ_α για την εκτέλεση ενός έργου είναι ο ίδιος για όλα τα έργα και εάν υπάρχουν πάντα αρκετοί επεξεργαστές διαθέσιμοι στο σύστημα, δηλαδή $p \geq B\Pi(t_k)$ για όλα τα k τότε $A\rho\iota\theta\ (n, p) = K$ και αυτό αποτελεί την ελάχιστη τιμή του $A\rho\iota\theta\ (n, p)$. Αν μάλιστα ο $B\Pi$ είναι σταθερός, τότε με $B_\kappa = p$ ανεξάρτητο του k, κάθε επεξεργαστής θα εκτελεί πάντα από μία πράξη σε κάθε επίπεδο και θα ισχύει βέβαια $n_{//} = Kp$, και $A\rho\iota\theta.(n, p) = n_{//}/P$.

Στην περίπτωση που ο πλεονασμός είναι $\rho = n_{//}/n = 1$ θα έχουμε $A\rho\iota\theta.(n, p)\tau_\alpha = T_{seq}(n)/P$, που είναι το ιδανικό ελάχιστο.

Οι επιβαρύνσεις όμως λόγω του $P>1$ μπορεί να είναι μέγιστες, μέγιστος αριθμός επεξεργαστών.

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ

Τότε γενικότερα ισχύει ότι οι ωφέλιμες πράξεις θα είναι:

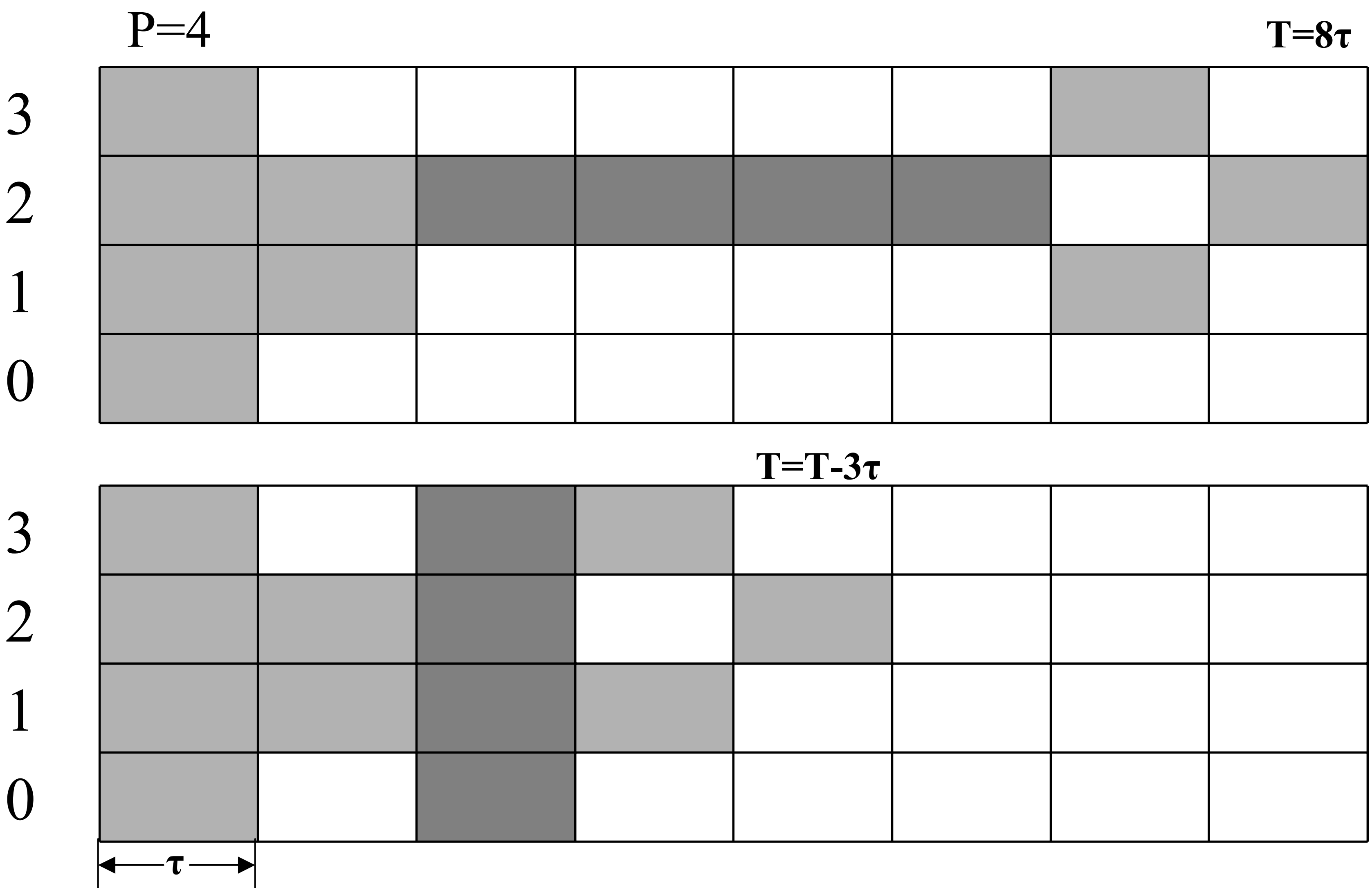
$$\Omega\,(n,\,p) = \sum_{k=0}^{K} \lceil B_k / p \rceil$$

αφού τα B_k έργα του ίδιου επιπέδου k θα εκτελεσθούν σε $\lceil B_k / P \rceil$ βήματα. Ενώ οι επιβαρύνσεις λόγω του P>1 συμβαίνουν μεταξύ των διαδοχικών φάσεων εκτέλεσης *(k, k+1)* και απαιτούν $\varepsilon_k(n,P)$ μονάδες επικοινωνίας τ_ε. Που είναι:

$$E\pi\iota\beta\,(n,\,p) = \sum_{k=0}^{K-1} \varepsilon_k(n,p)$$

ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ

Παράδειγμα κατανομής και εξισορρόπησης φορτίου



ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ
ΠΑΡΑΔΕΙΓΜΑ

Έστω το ακόλουθο το τμήμα ψευδοκώδικα:

```
PROGRAM (Segment)
.
.
.
A=B+C
B=D+E
X=Z+Y
B=B+Y
  DO I=1, 100
    S=(X*I)+(2*B)-((A+C)/(D-E))
  ENDDO
.
.
.
ENDPROGRAM
```

Πως μπορεί να γίνει η βέλτιστη κατανομή φορτίου για το συγκεκριμένο τμήμα κώδικα, αν υποθέσουμε ότι έχουμε στη διάθεση μας σύστημα με απεριόριστους υπολογιστικούς πόρους.

ΝΟΜΟΣ ΤΟΥ AMDAHL

Ο **Νόμος Amdahl** εκφράζεται από την ανισότητα

χρονοβελτίωση ≤ 1/(ποσοστό μη παραλληλοποιήσιμου τμήματος)

Δηλαδή: όλοι οι αλγόριθμοι έχουν έστω και πολύ μικρό μη παραλληλοποιήσιμο τμήμα (*α*). Ακόμα κι' αν αγνοηθούν οι χρόνοι επιβαρύνσεων, η χρονοβελτίωση δεν θα μπορεί να υπερβεί την τιμή *1/α*. Αν για παράδειγμα υπάρχει ένα 5% μη παραλληλοποιήσιμο τμήμα μίας εφαρμογής, θα ισχύει

χρονοβελτίωση ≤ 20

και η αύξηση του αριθμού των επεξεργαστών από κάποιο σημείο και μετά δεν θα βελτιώνει την χρονοβελτίωση.