Παράλληλοι Αλγόριθμοι

Στην ενότητα αυτή θα μελετηθούν τα εξής θέματα:

Το μοντέλο PRAM

Πολλαπλασιασμός πινάκων

Υπολογισμός αθροισμάτων προθέματος

Παράλληλοι Αλγόριθμοι

- Οι σύγχρονοι παράλληλοι υπολογιστές έχουν μεγάλα αποθέματα από επεξεργαστές, οι οποίοι συνεργάζονται για την επίλυση προβλημάτων. Για εκμετάλλευση αυτών των αποθεμάτων έχουν δημιουργηθεί παράλληλοι αλγόριθμοι, δηλαδή αλγόριθμοι οι οποίοι εκτελούν πολλές εντολές ταυτόχρονα.

- Μοντέλο επεξεργασίας PRAM (parallel random-access machine). Υπάρχουν
  - $p>1$ ανεξάρτητοι επεξεργαστές ο καθένας από τους οποίους έχει το δικό του "πρόγραμμα", και
  - μια μεγάλη κοινόχρηστη μνήμη.
  - Όλοι οι επεξεργαστές μπορούν να διαβάζουν και να γράφουν πληροφορίες από και προς την κοινόχρηστη μνήμη ταυτόχρονα. Επίσης μπορούν να εκτελούν πράξεις ταυτόχρονα.

Παράλληλοι Αλγόριθμοι

- Διαχωρίζουμε τους PRAM αλγόριθμους στις εξής κατηγορίες:
  1. EREW, exclusive read, exclusive write.
  2. CREW, concurrent read, exclusive write.
  3. ERCW, exclusive read, concurrent write.
  4. CRCW, concurrent read, concurrent write.

  ανάλογα με το αν γράφουν/διαβάζουν την ίδια κυψελίδα μνήμης ταυτόχρονα.

- Σχεδιάζοντας παράλληλους αλγόριθμους, ο στόχος είναι η ελαχιστοποίηση του αριθμού των παράλληλων βημάτων. Ο αριθμός αυτός παριστάνει το χρόνο εκτέλεσης του αλγορίθμου.

- Θα θέλαμε οι παράλληλοι αλγόριθμοι που σχεδιάζουμε να είναι δραματικά γρηγορότεροι από τους γρηγορότερους ακολουθιακούς αλγόριθμους για το ίδιο πρόβλημα. Επίσης οι αλγόριθμοί μας δεν θα πρέπει να χρησιμοποιούν "παράλογα" πολλούς επεξεργαστές.

Πολλαπλασιασμός πινάκων

- Το πρόβλημα πολλαπλασιασμού πινάκων αποτελεί παράδειγμα προβλήματος όπου ο προφανής ακολουθιακός αλγόριθμος μπορεί να παραλληλοποιηθεί κατ΄ευθείαν.

- Μας δίδονται δύο n×n πίνακες, A και B, και ζητείται το γινόμενο C=A·B, δηλαδή τα n^2 στοιχεία

$$C[i, j] = \sum_{k=1}^{n} A[i,k] \cdot B[k, j]$$

$1 \le i,j \le n$.

Προσπάθεια 1

- Εξάγουμε όσο περισσότερο "παραλληλισμό" μπορούμε από τον προφανή ακολουθιακό αλγόριθμο.

- Όλα τα n^3 γινόμενα A[i,k]·B[k,j], $1 \le i,j,k \le n$, μπορούν να υπολογισθούν σε χρόνο 1 *ανεξάρτητα* από n^3 επεξεργαστές.

Πολλαπλασιασμός πινάκων

- Ονομάζουμε ⟨i,k,j⟩ τον επεξεργαστή που υπολογίζει το γινόμενο A[i,k]·B[k,j].
- Στη συνέχεια, n^2 από τους επεξεργαστές, έστω οι επεξεργαστές ⟨i,1,j⟩, συλλέγουν τα n γινόμενα που αντιστοιχούν στην ίδια συνιστώσα C[i,j], και τα προσθέτουν σε n-1 επιπρόσθετα παράλληλα βήματα.
- *Ο συνολικός χρόνος είναι n και ο συνολικός αριθμός επεξεργαστών είναι n^3.*
- Η ελάττωση του χρόνου εκτέλεσης από n^3 (για τον ακολουθιακό αλγόριθμο) σε n (για τον παράλληλο αλγόριθμο) είναι βέβαια σημαντική, όχι όμως του είδους της δραματικής βελτίωσης για την οποία θα άξιζε να κατασκευάσουμε παράλληλους υπολογιστές.
- Αυτό που θα θέλαμε να δούμε είναι η εκθετική πτώση του χρόνου εκτέλεσης ακολουθιακών αλγόριθμων. Για πολυωνυμικούς ακολουθιακούς αλγόριθμους θα θέλαμε να έχουμε αντίστοιχους πολυλογαριθμικούς παράλληλους αλγόριθμους.

Πολλαπλασιασμός Πινάκων

Προσπάθεια 2

- Αντί να αναθέσουμε όλες τις προσθέσεις στον ίδιο επεξεργαστή, οργανώνουμε τους επεξεργαστές έτσι ώστε να εκτελέσουν ένα δυαδικό δέντρο από προσθέσεις. Με αυτό τον τρόπο μπορούμε να υπολογίσουμε όλα τα αθροίσματα μόνο με lg n παράλληλα βήματα.

- Πιο συγκεκριμένα:

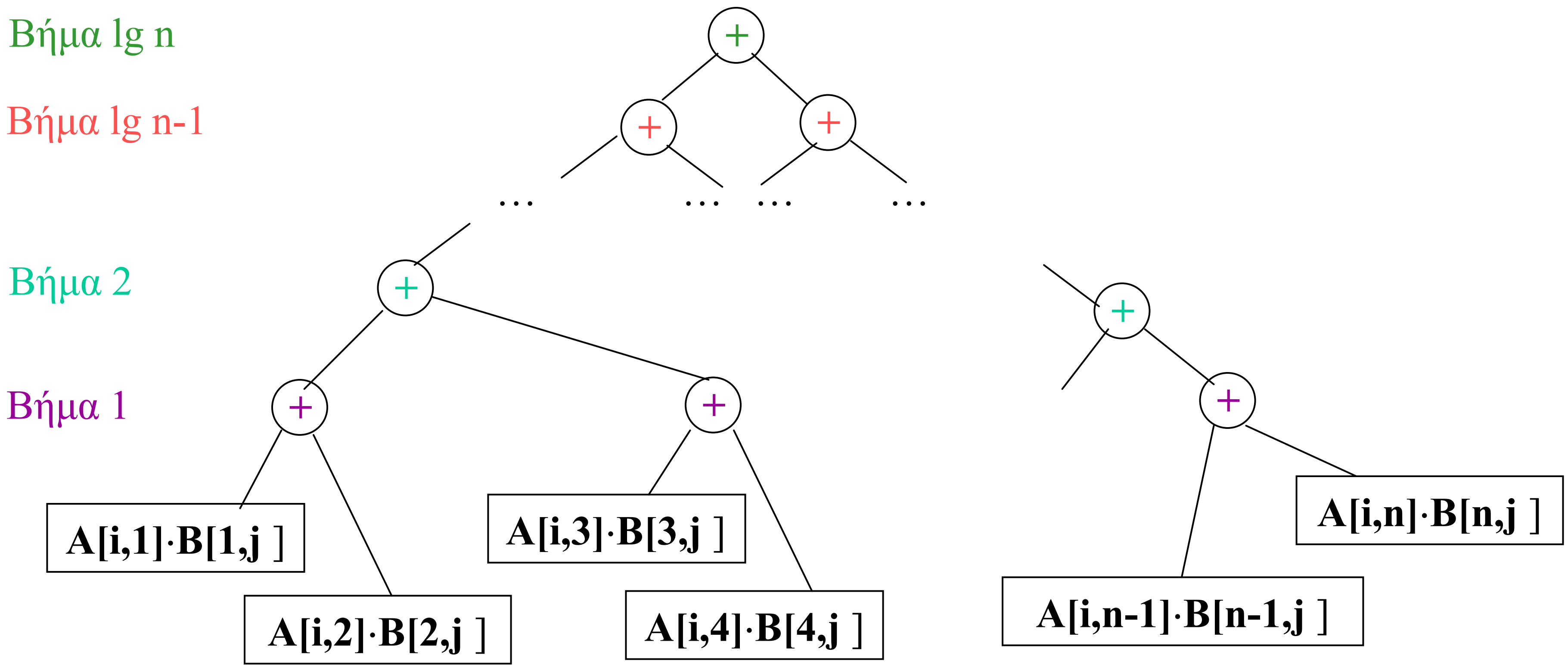
  Για τον υπολογισμό της συνιστώσας C[i,j] συμμετέχουν όλοι οι επεξεργαστές

  $\langle i,1,j\rangle, \langle i,2,j\rangle, \ldots \langle i,n,j\rangle$

  και όχι μόνο ο $\langle i,1,j\rangle$ όπως πριν.

- Υποθέτουμε χωρίς βλάβη της γενικότητας ότι το n είναι δύναμη του 2.

Προσθέτοντας παράλληλα



- Υπάρχουν lg n παράλληλα βήματα και ο κάθε επεξεργαστής έχει ένα καταχωρητή στη διάθεσή του.

Προσθέτοντας παράλληλα

- Στο παράλληλο βήμα 1, οι επεξεργαστές $\langle i,2,j\rangle$, $\langle i,4,j\rangle$, …, $\langle i,n,j\rangle$ υπολογίζουν αθροίσματα ως εξής

$\langle i,2,j\rangle$	$A[i,1]\cdot B[1,j] + A[i,2], B[2,j]$
$\langle i,4,j\rangle$	$A[i,3]\cdot B[3,j] + A[i,4], B[4,j]$
…	
$\langle i,n,j\rangle$	$A[i,n-1]\cdot B[n-1,j] + A[i,n], B[n,j]$

- Στο παράλληλο βήμα 2, οι επεξεργαστές $\langle i,4,j\rangle$, $\langle i,8,j\rangle$, …, $\langle i,n,j\rangle$ υπολογίζουν αθροίσματα ως εξής

$\langle i,4,j\rangle$	έξοδος του $\langle i,2,j\rangle$ στο βήμα 1 + έξοδος του $\langle i,4,j\rangle$ στο βήμα 1
$\langle i,8,j\rangle$	έξοδος του $\langle i,6,j\rangle$ στο βήμα 1 + έξοδος του $\langle i,8,j\rangle$ στο βήμα 1
…	

- Τελικά στο παράλληλο βήμα lg n, ο επεξεργαστής $\langle i,n,j\rangle$ υπολογίζει το συνολικό άθροισμα

$$C[i,j] = \sum_{k=1}^{n} A[i,k]\cdot B[k,j]$$

Χρόνος Εκτέλεσης

- Ο συνολικός αριθμός των παράλληλων βημάτων είναι $1 + \lg n$

για τον υπολογισμό των n^3 γινομένων $A[i,k] \cdot B[k,j]$

για τον υπολογισμό των n^2 αθροισμάτων $C[i,j] = \sum_{k=1}^{n} A[i,k] \cdot B[k,j]$

- Πόσο είναι το συνολικό έργο (ο συνολικός αριθμός αριθμητικών πράξεων;)

$$n^3 + n^2\left(\frac{n}{2} + \frac{n}{4} + \ldots + 1\right) = n^3 + n^3\left(\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^{\lg n}}\right)$$

$$= n^3\left(1 + \frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^{\lg n}}\right)$$

$$= n^3\left(\frac{1-(1/2)^{\lg n+1}}{1-1/2}\right) = 2n^3(1 - 1/2n)$$

$$= 2n^3 - n^2$$

- Πόσο είναι το συνολικό έργο του αντίστοιχου ακολουθιακού αλγόριθμου;

$n^2(n+n-1) = n^2(2n-1) = 2n^3 - n^2$

Υπολογισμός αθροισμάτων προθέματος

- Στο πρόβλημα υπολογισμού αθροισμάτων προθέματος έχουμε σαν είσοδο n αριθμούς x_1, x_2,…, x_n και θέλουμε να υπολογίσουμε όλα τα προθεματικά αθροίσματα
- Υποθέτουμε ότι το n είναι δύναμη του 2.
- Έχουμε τον πιο κάτω αναδρομικό αλγόριθμο:

```
ParallelSum(x[1,…,n])
  παράλληλα υπολόγισε τα αθροίσματα
    y1=x1+x2, y2=x3+x4, … , yn/2=xn-1+xn
  Z=ParallelSum(y[1,…,n/2]);
  παράλληλα επέστρεψε
     x1       ,        Z[1],
    Z[1]+ x3,        Z[2]
    Z[2]+ x5,        Z[3]
    …
    Z[n/2-1]+ xn-1, Z[n/2]
```

Ορθότητα

- Το τελευταίο βήμα της διαδικασίας, παίρνει τα n/2 προθεματικά αθροίσματα της ακολουθίας

  $x_1+x_2, x_3+x_4, \ldots , x_{n-1}+x_n$

  δηλαδή, τα

  x_1+x_2

  $x_1+x_2 + x_3+x_4$

  $x_1+x_2 + x_3+x_4 + x_5+x_6$

  …

  $x_1+x_2 + x_3+x_4 + \ldots + x_{n-1}+x_n$

  και τα επιστρέφει μαζί με τα υπόλοιπα προθεματικά αθροίσματα της ακολουθίας $x_1, x_2, x_3, \ldots , x_n$ τα οποία λαμβάνονται προσθέτοντας $x_3, x_5, x_7, \ldots, x_{n-1}$ αντίστοιχα στα αθροίσματα αυτά:

  $x_1+x_2 \qquad + x_3$

  $x_1+x_2 + x_3+x_4 \qquad + x_5$

  $x_1+x_2 + x_3+x_4 + x_5+x_6 \qquad + x_7$

  …

  $x_1+x_2 + x_3+x_4 + \ldots + x_{n-3}+x_{n-2} \qquad + x_{n-1}$

Χρόνος Εκτέλεσης

- Ποιος είναι ο συνολικός αριθμός παράλληλων βημάτων της αναδρομικής διαδικασίας;

- Από την αναδρομική κατασκευή του έχουμε

$$
\begin{aligned}
T(n) &= T(n/2) + 2 \\
&= T(n/4) + 2 + 2 \\
&= \ldots \\
&= T(1) + 2 + \ldots + 2 + 2 \\
&= 2 \lg n
\end{aligned}
$$