PASSO INDUTTIVO: supponendo vera la (*) per $j-1$ dimostro che vale per j

istruzione del prog. → $M[2^j k] = M[2^j k] + M[2^j k - 2^{j-1}]$

$M[2^{j-1} 2k] \underset{(*)}{=} M[2^{j-1} 2k] + \ldots + M[2^{j-1}(2k-1)+1]$

successivo

precedente

$M[2^{j-1}(2k-1)] \underset{(*)}{=} M[2^{j-1}(2k-1)] + \ldots + M[2^{j-1}(2k-2)+1]$

$2^j(k-1)+1$

$M[2^j k] = M[2^j k] + \ldots + M[2^j(k-1)+1]$

↓

la (*) è vera!

VALUTAZIONE DELL'ALGORITMO SOMMATORIA:

$P(n) = \frac{n}{2}$

$T(n, \frac{n}{2}) = \ldots$ microistruzioni: LD, LD, ADD, ST

$= 4 \log n$

E se n non è potenza di 2?

Idea: allunghiamo l'input di 0 fino alla potenza di 2 successiva ad n.

cioè da n $\xrightarrow{\text{binario}}$ $1\, b_t\, b_{t-1} \ldots b_0$

$1\,0\,0\,0 \ldots 0$

a $2n$ $\xrightarrow{\text{binario}}$ $1\, b_t\, b_{t-1} \ldots b_0\, 0$

Pertanto avrò:

$$p(n) = \frac{2n}{2} = n$$

$$T(n,n) = 4 \log 2n \leq 5 \log n$$

$$\Rightarrow \quad p(n) = O(n) \quad e \quad T(n,n) = O(\log n)$$

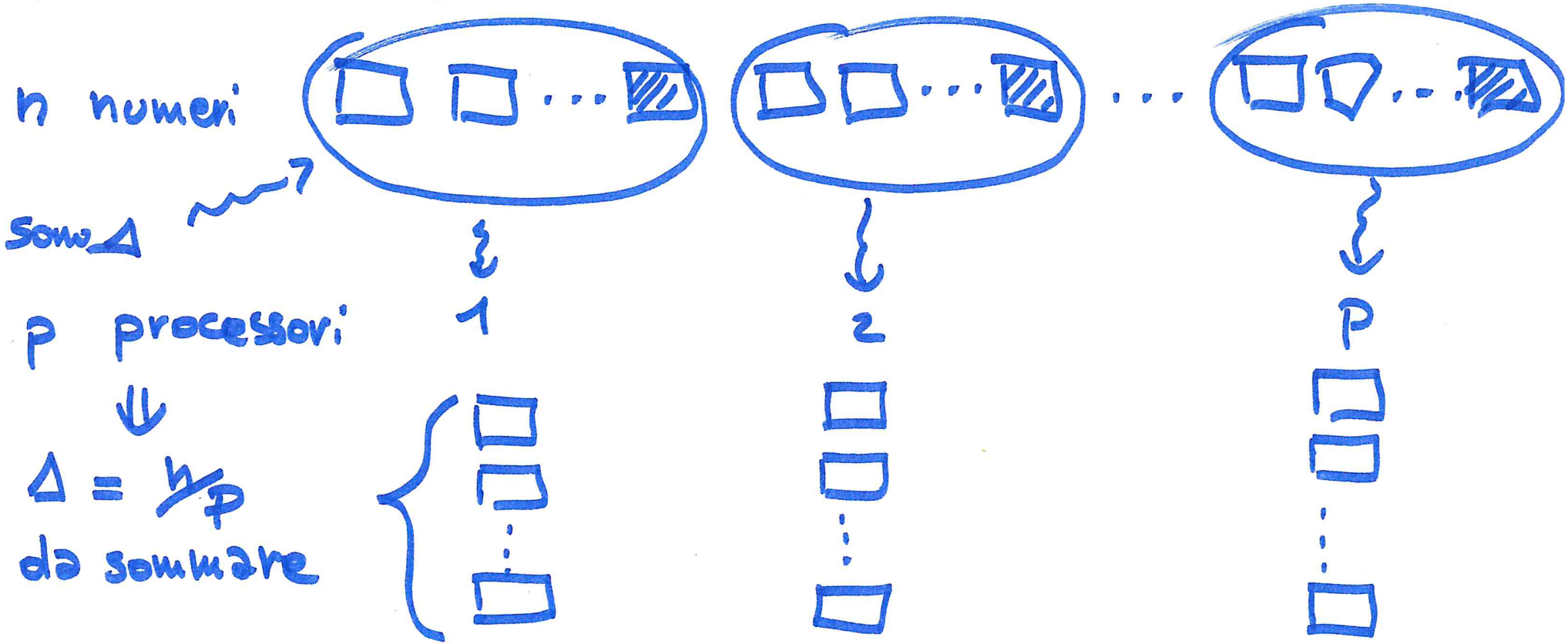
EFFICIENZA

$$E(n,n) = \frac{n-1}{n \; 5 \log n} \sim \frac{n}{5 \log n} \to 0$$ lentamente

$\Rightarrow$ visto che i processori sono un po' sprecati (vedi il primo passo) applichiamo Wyllie:

$p(n) = o(n)$ per avere $E \to k \neq 0$

Applichiamo Wyllie



1° passo parallelo

$1 \le k \le p \qquad M[k\Delta] = M[k\Delta] + \dots + M[(k-1)\Delta+1]$

passi paralleli successivi

Uso l'algoritmo per SOMMATORIA visto prima su

$$M[\Delta],\ M[2\Delta],\ \dots,\ M[p\Delta]$$

che memorizza in $M[p\Delta] = M[n] = \sum_i M[i]$

CORRETTEZZA: O.K.

VALUTAZIONE:

$p(n) = p$

$$T(n,p) = T(1^{\text{o}}\text{passo}) + T(\text{passi succ.}) = \frac{n}{p} + 5\log p$$

$$E(n,p) = \frac{n-1}{p\left(\frac{n}{p} + 5\log p\right)} > \frac{n-1}{n + \underbrace{5p\log p}_{n}} \to k \neq 0$$

$$\Rightarrow E \approx \frac{n}{2n} \to \frac{1}{2}$$

Vorrei $p \log p = \frac{n}{5} \rightsquigarrow p = \frac{n}{\log n \cdot 5}$

Infatti: $\frac{n}{5 \log n}\left(\log \frac{n}{(\log n)5}\right) = \frac{n}{5 \log n}\left(\log n - \log 5 - \log \log n\right)$

$$= \frac{n}{5}\left(1 - \underbrace{\frac{\log 5}{\log n}}_{\to 0} - \underbrace{\frac{\log \log n}{\log n}}_{\to 0}\right) \sim \frac{n}{5}$$

Ricapitolando:

$$p(n) = \frac{n}{5 \log n}$$

$$T(n, p(n)) = 5 \log n + 5 \log n - \ldots \leq 10 \log n$$

[Uso un numero di processori sublineare e mantengo un tempo logaritmico] Wyllie

Domanda: possiamo fare meglio per SOMMATORIA?

LOWER BOUND (idea)

- per SOMMATORIA possiamo visualizzare un algo parallelo usando un albero:

  foglie = numeri da som. livelli = passi paralleli

  Livello con più nodi = numero di processori

  altezza dell'albero = Tempo dell'algo

  ES: Il primo albero = algoritmo sequenziale

  Il secondo albero = " parallelo

- Un qualunque albero con n numeri ha n foglie

  $n = \#$ di foglie $\leq 2^h$ $\Rightarrow h \geq \log n$

  con h = altezza

  h ↓ TEMPO

SOMMATORIA COME SCHEMA PER ALTRI PROBLEMI

OP iterata dove op è associativa

Input: $M[1], \dots, M[n]$

Output: $\underset{i}{Op}\ M[i] \to M[n]$

Es: $op = +, *, \wedge, \vee, \oplus, \min, \max, \text{“}\cdot\text{”}, \dots$

Abbiamo una soluzione efficiente parallela

$$P = O\left(\frac{n}{\log n}\right) \qquad T = O(\log n)$$

In realtà con modelli di PRAM + potenti posso ottenere Tempo costante $\Rightarrow$

CRCW - PRAM

Problema $\wedge$-iterato : $M[n] = \bigwedge_i M[i]$

Programma

for $1 \leq k \leq n$ pardo

if $M[k] = 0$ Then

$M[n] = 0$

NOTA : CW con politica COMMON

$p(n) = n$

$T(n,n) = 3$

$E(n,n) = \frac{n-1}{n \cdot 3} \to \frac{1}{3}$

SOMMATORIA COME SOTTOPROBLEMA DI ALTRI

- prodotto interno di vettori : $\langle \ldots, \ldots \rangle$
- prodotto matrice vettore
- prodotto matrice matrice
- potenza di una matrice