Broadcast su P-RAM EREW

```
Broadcast (x)
{
P0:   A[0] = x
      for i=0 to log n - 1 do
          for j = 2^i to 2^(i+1) - 1 pardo
              A[j] = A[j - 2^i]
      Pj:
}
```

$t(n, n) = O(\log n)$

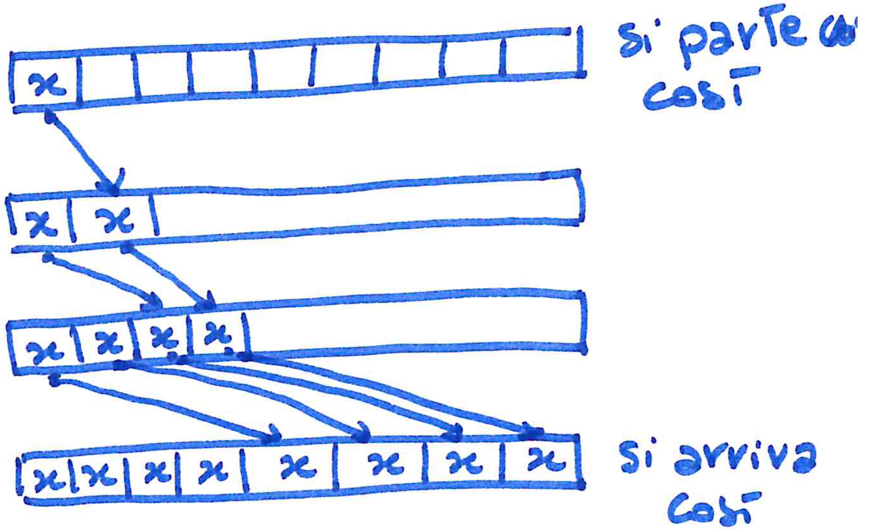
$i = 0 \quad j \in [1, 1]$

$i = 1 \quad j \in [2, 3]$

$i = 2 \quad j \in [4, 7]$

$\vdots$

indici del for pardo



Uso di Broadcast su P-RAM

```
Cerca ( A, n, x )
{
    Broadcast (x)
    indice = -1
    for i=0 to n-1 pardo
Pi:     if A[i] = x[i] then indice = i
    return indice
}
```

- $t(n,n) = O(\log n)$
- se A[i] tutti distinti ⇒ EREW
- altrimenti ⇒ ERCW
- ERCW può essere trasformato in EREW (con un aumento della funzione tempo)

EFFICIENZA

Un primo confronto tra i tempi

$$T(n,1) \longleftrightarrow T(n, p(n))$$

Abbiamo 2 casi

$$T(n, p(n)) = \Theta(T(n,1))$$

NO

$$T(n, p(n)) = o(T(n,1))$$

SI

Speed - UP

$$S(n, p(n)) = \frac{T(n,1)}{T(n, p(n))}$$

Esempio: $S = 4$ l'algoritmo parallelo è 4 volte + veloce del sequenziale

Ponendoci nel caso: $T(n,p(n)) = o(T(n,1))$

$S(n, p(n)) \to \infty$

Domanda:
stiamo considerando $p(n)$?

Risposta:
NO

Esempio di problema:

Soddisfacibilità di formule

(dove la lunghezza della formula è legata linearmente al numero di variabili coinvolte)

$T(n,1) = 2^n$ per il momento ($\in NP$)

$T(n, p(n)) = \Theta(n)$

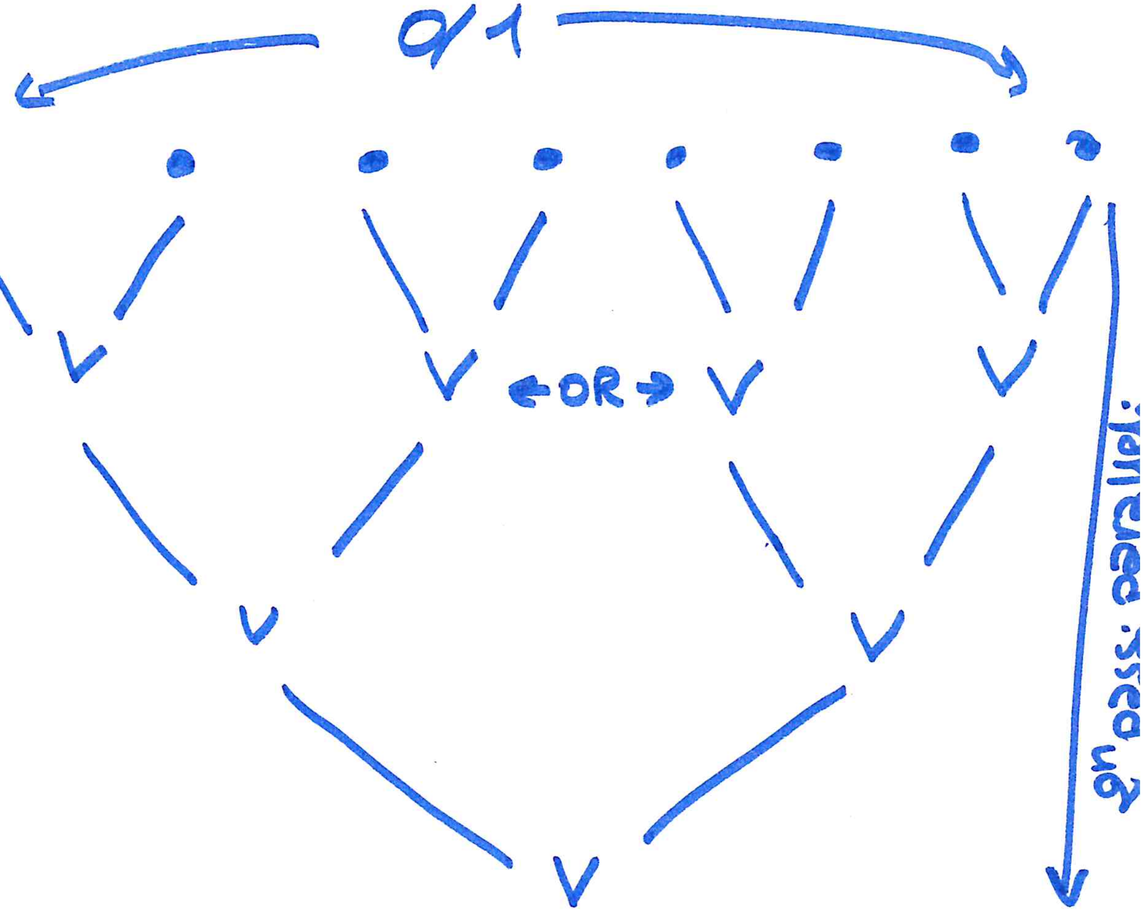
STRATEGIA
utilizzo 1 processore per ogni assegnamento

$S(n, p(n)) = \frac{2^n}{n} \to \infty$

infinito ma $p(n) = 2^n$

Soddisfacibilità di formule F

Assegnameni 2^n per n variabili



$\Rightarrow$ 2^n processori rispondono:

1 se F viene soddisfatta

0 se F non viene soddisf.

n° di passi paralleli

$\log 2^n = n$

$\boxed{\text{Efficienza}}$

$$E(n, p(n)) = \frac{S(n, p(n))}{p(n)} = \frac{T(n,1)^*}{p(n) \cdot T(n, p(n))}$$

* si intende il tempo del miglior algoritmo sequenziale (o lower bound)

<u>Domanda</u>:

Vedremo che per il parametro E vale:

$$0 \leq E(n, p(n)) \leq 1$$

Esempio: soddisfacibilità di formule

$$E(n, p(n)) = \frac{\cancel{2^n}}{\cancel{2^n}\, n} = \frac{1}{n} \to 0$$

OSSERVAZIONE

Quando $E \to 0$, stiamo usando troppi processori che magari vengono inutilizzati per la maggioranza del tempo.

<u>Perchè ?</u>

Vediamo $E \leq 1$ in due modi diversi.

<u>I modo</u>: attraverso Algo parallelo $\rightarrow$ Algo sequenz.

	P_1	P_2	$\dots$	$P(n)$	
1° passo parallelo	$\square^{1}$	$\square^{2}$		$\square^{P(n)}$	$t_1(n)$
2° passo	$\square^{P(n)+1}$	$\square^{P(n)+2}$		$\square^{\dots}$	$t_2(n)$
$\dots$					
$k(n)$° passo	$\square$	$\square$		$\square^{\dots}$	$t_{k(n)}(n)$

La trasformazione: sequenzializzo i passi paralleli

Tempo: $\tilde{T}(n,1)$

$$\underline{T(n,1)} \leq \tilde{T}(n,1) \leq p(n)\,t_1(n) + p(n)\,t_2(n) + \dots + p(n)\,t_{k(n)}(n) =$$

$$= p(n) \sum_{i=1}^{k(n)} t_i(n)$$

$$= \underline{p(n)\,T(n, p(n))}$$

Si ottiene

$$T(n,1) \leq p(n) \cdot T(n, p(n)) \qquad (*)$$

Osservazione a latere: da $(*)$ ricavo che

$$\frac{T(n,1)}{p(n)} \leq T(n, p(n))$$

se $T(n,1)$ è il tempo sequenziale ottimo
$\Rightarrow$ il meglio che posso fare con un algoritmo parallelo è distribuire equamente tra i processori il lavoro del sequenziale

Ancora da $(*)$ si ottiene:

$$\underbrace{\frac{T(n,1)}{p(n)\,T(n,p(n))}}_{E(n,\,p(n))} \leq 1$$

<u>Commentiamo</u> il risultato $0 \leq E \leq 1$

- Se $E \to 0$ non va bene:
  Infatti: dato che $T(n,p(n)) = o\,(T(n,1))$
  deve essere che $p(n)$ cresce troppo velocemente

$\Rightarrow$ Il meglio che possiamo avere è

$E \to K \leq 1$ dove K è una COSTANTE

II modo: (per $E \leq 1$)

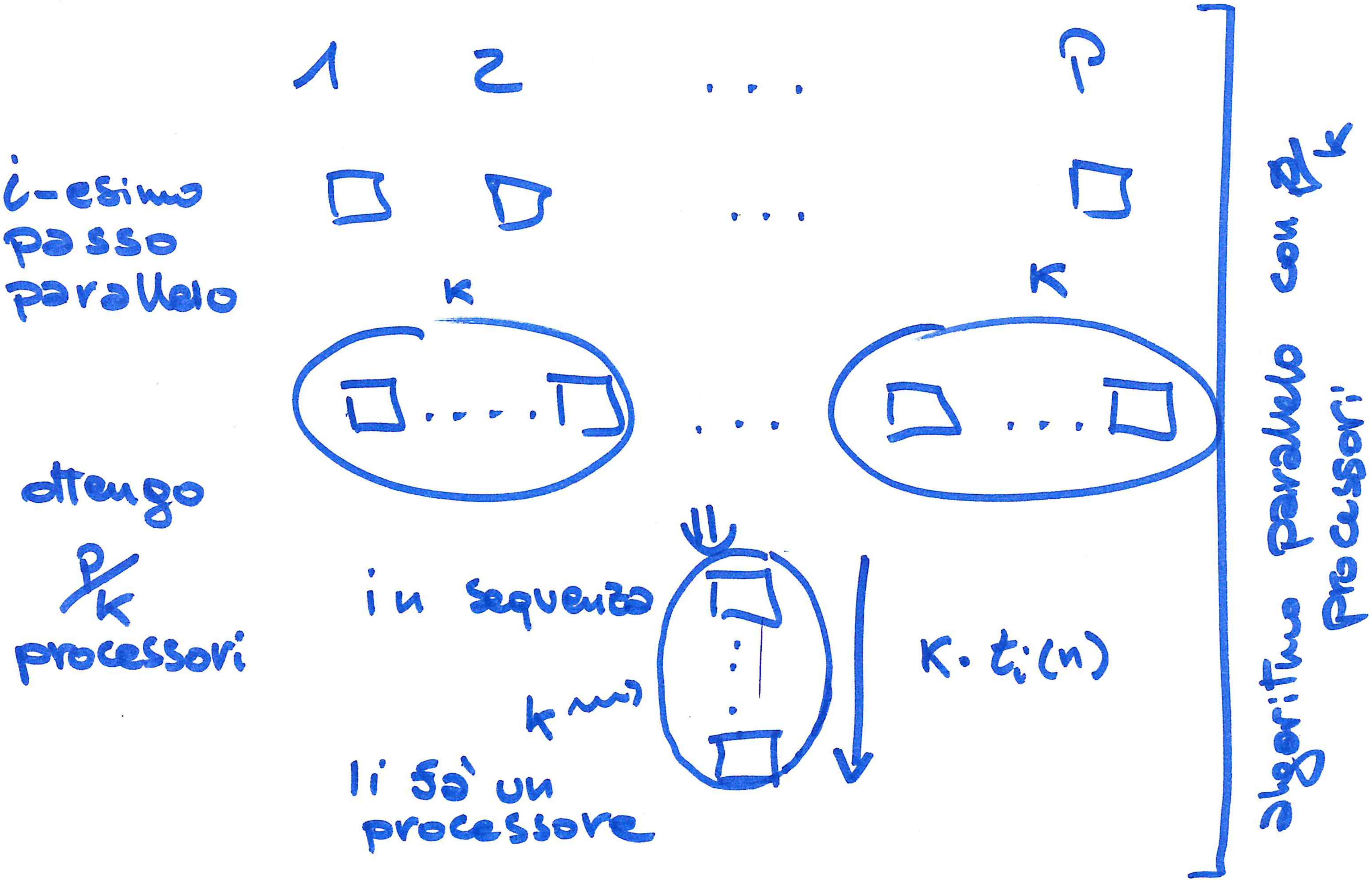
<u>Ricetta</u> : [J. Wyllie 1979 PhD Thesis]

Se $E \to 0$ allora per migliorare l'algoritmo prova a ridurre $p(n)$ senza degradare il tempo

Vediamo la validità di questo principio cambiando il numero di processori

$$\text{da} \quad p \longrightarrow \text{a} \quad p/k$$

Modifica dell'algoritmo parallelo



Tempo ?

$$T(n, p/k) \leq \sum_{i=1}^{k(n)} k \cdot t_i(n)$$

$$= k \cdot \sum_{i=1}^{k(n)} t_i(n)$$

$$= k \cdot T(n, p)$$

si ha: $T(n, p/k) \leq k \cdot T(n, p)$

Provate a far vedere che la E cresce
diminuendo i processori

$$E(n, p/k) \longleftrightarrow E(n, p)$$

$\Rightarrow$ DECRESCE CON L'AUMENTARE DI P