VALUTAZIONE DELL'ALGORITMO POINTER DOUBLING:

$p(n) = n-1$

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

- $M[S[k]] = M[k] + M[S[k]]$

$$5 \left\{ \begin{array}{ll} \text{LOAD} & M[k] \\ \text{LOAD} & S[k] \\ \text{LOAD} & M[S[k]] \\ \text{ADD} & \\ \text{STORE} & M[S[k]] \end{array} \right.$$

- $S[k] = (S[k] == 0\,?\ 0 : S[S[k]])$

$$4 \left\{ \begin{array}{ll} \text{LOAD} & S[k] \\ \text{JZERO} & \\ \text{LOAD} & S[S[k]] \\ \text{STORE} & S[k] \end{array} \right.$$

$J = 1, \dots, \log n$

$\downarrow$

$T(n, n-1) \sim 9 \log n$

$$E(n, p(n)) = \frac{n-1}{(n-1)\, 9 \log n} = \frac{1}{9 \log n} \to 0 \text{ lentamente}$$

Sfruttiamo Wyllie come per SOMMATORIA
(in modo da far sparire la funzione $\log n$ da E)

$$\Rightarrow \quad p(n) = O\left(\frac{n}{\log n}\right) \quad T(n, p(n)) = O(\log n) \quad E \to C \neq 0$$

come per SOMMATORIA anche l'algoritmo dato per
SOMME-PREFISSE può essere usato per:

OP-PREFISSA

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

Output: $M[k] = \mathop{OP}_{i=1}^{k} M[i]$, $1 \leq k \leq n$

OP deve essere associativa come ad es:

$+$, $*$, $\wedge$, $\vee$, min, max, "$\cdot$", ...

NUOVO PROBLEMA: valutazione di polinomi

INPUT: $p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$, α

OUTPUT: $p(\alpha)$

Dati in memoria M:

- il valore α
- $a_0, a_1, \dots, a_n \longrightarrow A[0], A[1], \dots, A[n]$

Algoritmo Tradizionale sequenziale:

prodotti: $\sum_{i=0}^{n} i \sim n^2$

somme: n

$\Big] + \sim n^2$

Miglioramento di <u>Ruffini - Horner</u>

idea : $p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$

$\downarrow$

raccogli x in maniera iterata

$$= a_0 + x(a_1 + a_2 x + a_3 x^2 + a_4 x^3)$$

$$= a_0 + x(a_1 + x(a_2 + a_3 x + a_4 x^2))$$

$$= a_0 + x(a_1 + x(a_2 + x(a_3 + a_4 x)))$$

generalizziamo :

$$p(x) = a_0 + x(a_1 + \ldots \quad \underbrace{a_{n-2} + \underset{+\,\alpha}{x}(\underbrace{a_{n-1} + \overset{p}{a_n}\underset{\alpha}{x}}_{p})\ldots)}_{p}$$

$$P = a_j + p \cdot \alpha$$

Codice per algo. seq. Ruffini Horner

Input (α)
$p = a_n$
for $i = 1$ to n
$p = a_{n-i} + p\alpha$
Output (p)

prestazioni:

$T(n,1) = 2n$

<u>Possibile algoritmo parallelo</u>

(1) - costruisco il vettore delle potenze di α : Q

$Q[k] = \alpha^k \qquad 0 \leq k \leq n$

- eseguo il prodotto interno $\langle A, Q \rangle$

$$\langle A, Q \rangle = \sum_{k=0}^{n} A[k] \cdot Q[k]$$

- restituisco $\langle A, Q \rangle$

Per risolvere il punto (1):

- metto α in tutti gli elementi di Q da 1 a n

  $Q[1]=\alpha, \; Q[2]=\alpha, \; \dots \;, \; Q[n]=\alpha$

  si richiede di RISOLVERE REPLICA ...

- applico il PRODOTTO-PREFISSO su Q:

  $Q[1]=\alpha, \; Q[2]=\alpha^2, \; \dots \;, \; Q[n]=\alpha^n$

Come risolvere REPLICA in parallelo

- primo:

```
for k=1 to n par do
    Q[k]=α;
```

← essendo α in una cella di M è un accesso simul.

CREW

prestazioni

$p=n, \; t=2, \; E \sim \frac{n}{n \cdot 2} \to c \neq 0$

NOTA: se REPLICA è un MODULO DA USARE FORSE P=n è Troppo

Per abbassare il n° di processori di REPLICA $\Rightarrow$ Wyllie

Raggruppo gli n processori in $\log n$ elementi

Il k-esimo processore carica α nelle celle di posiz:

$$(k-1)\log n + 1, \dots, k \log n$$

- <u>secondo</u>:

for $k = 1$ to $n/\log n$ par do

for $i = 1$ to $\log n$ do

$Q[(k-1)\log n + i] = \alpha$ $\leftarrow$ CREW

prestazioni:

$$p = \frac{n}{\log n}, \quad t = c \log n, \quad E = \frac{n}{\frac{n}{\log n} c \cdot \log n} = \frac{1}{c} \neq 0$$

COSTANTE

- <u>Terzo</u> : desideriamo un <u>EREW</u> - PRAM

<u>1</u> costruisci il vettore $\alpha, 0, 0, \dots, 0$
<u>2</u> esegui SOMME-PREFISSE

Codice per ottenere : $\alpha, 0, 0, \dots, 0$

Input (α)
Q[1] = α
for K = 2 To n par do
    Q[k] = 0 ⟵ zero è una costante che non ha bisogno di essere letta

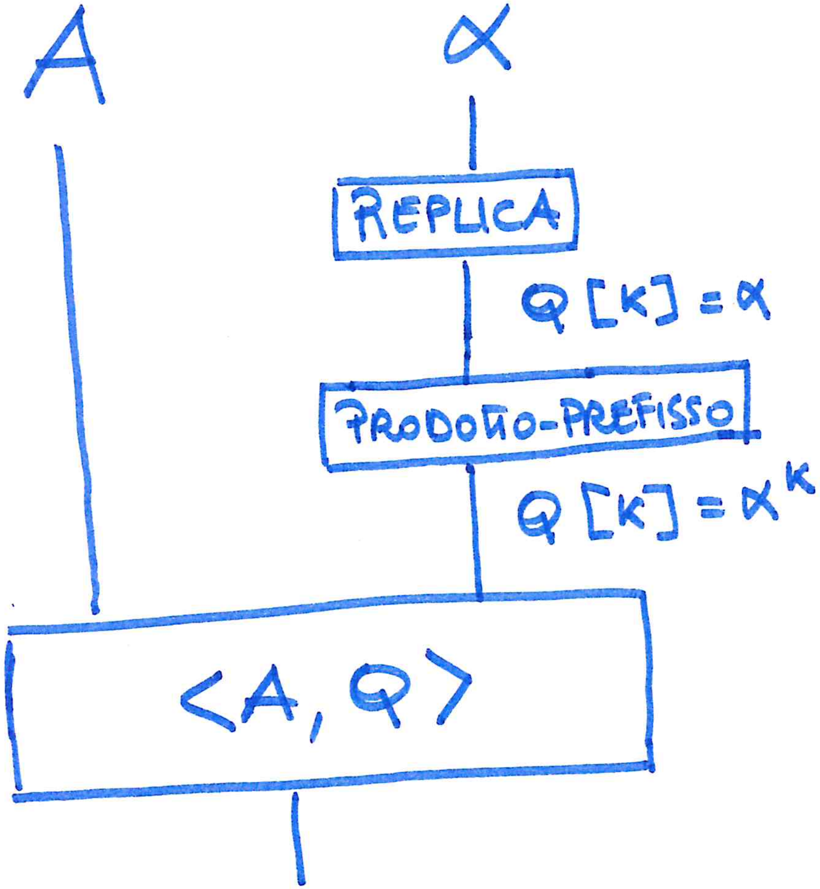
- <u>quarto</u> : riduzione dei processori con wyllie

prestazioni : <u>1</u> $p = n/\log n \quad t = \log n$

<u>2</u> $p = n/\log n \quad t = \log n$

TOT $p = n/\log n \quad t = \log n \Rightarrow E = \underset{\text{COSTANTE}}{C} \neq 0$

Riassunto: VALUTAZIONE POLINOMIO con $\boxed{\text{EREW PRAM}}$



p	t
$n/\log n$	$\log n$
$n/\log n$	$\log n$
$n/\log n$	$\log n$
$n/\log n$	$\log n$

$$E = \frac{T(n,1)}{p(n)\, T(n, p(n))} \sim \frac{2n}{\frac{n}{\log n} \log n} \rightarrow C \neq \text{COSTANTE}$$