L'array S è utile per risolvere 2 problemi:

① Attraversamento dell'albero in preordine

② Calcolare la profondità dei nodi

Servono due definizioni:

① $\forall v \in V$: $N(v)$ = ordine di attraversamento di v in preordine

② $\forall v \in V$: $P(v)$ = profondità di v nell'albero

Esempi:

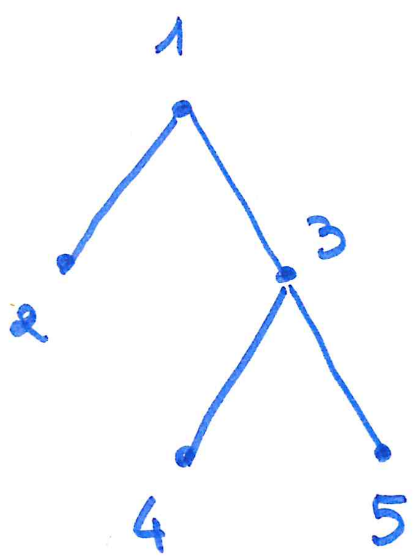
① $N(\text{radice}) = 1$ mentre $N(\text{foglia tutta a destra}) = n$

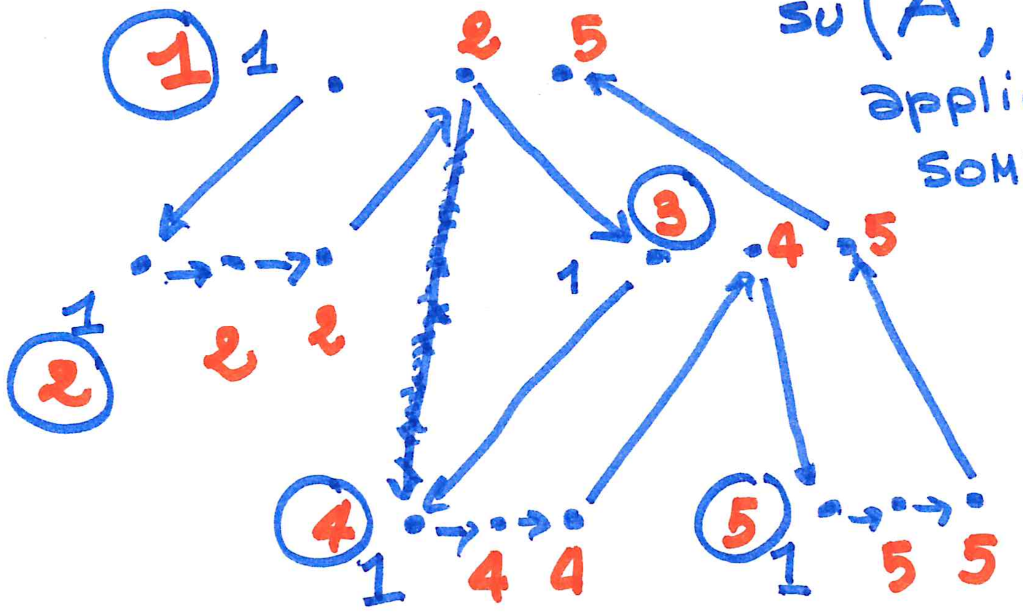
② $P(\text{radice}) = \cancel{1}\, 0$ " $P(\text{figlio radice}) = \cancel{2}\, 1$

Attraversamento dell'albero in Preordine

Definiamo un particolare array A:

$$A[(v,x)] = \begin{cases} 1 & \text{se } x = s \\ 0 & \text{se } x = c/d \end{cases}$$





su (A, S) applichiamo SOMME-PREFISSI

$$A[(v,s)] = N(v)$$

Algoritmo parallelo per l'ordine $N(v)$

~~Prestazioni~~ :

1° Calcolo di A e S (successore)

2° Calcolo somme-prefisse su A e S'

OUTPUT: è nel modo $A[(v, s)]$

Prestazioni: è EREW

$$p(n) = \frac{n}{\log n} \qquad T(n, p(n)) = \log n$$

sia 1°

che 2°

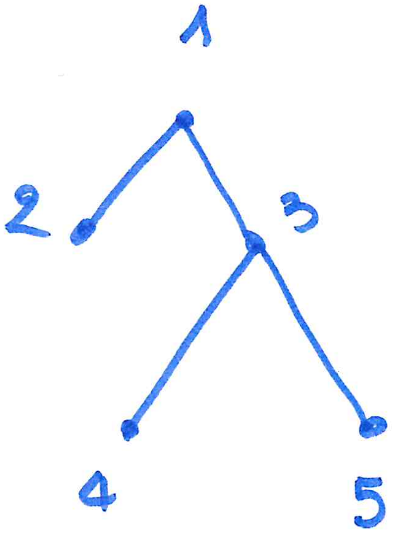
$$E = \frac{n}{\frac{n}{\log n} \cdot \log n} \rightarrow C \neq 0$$

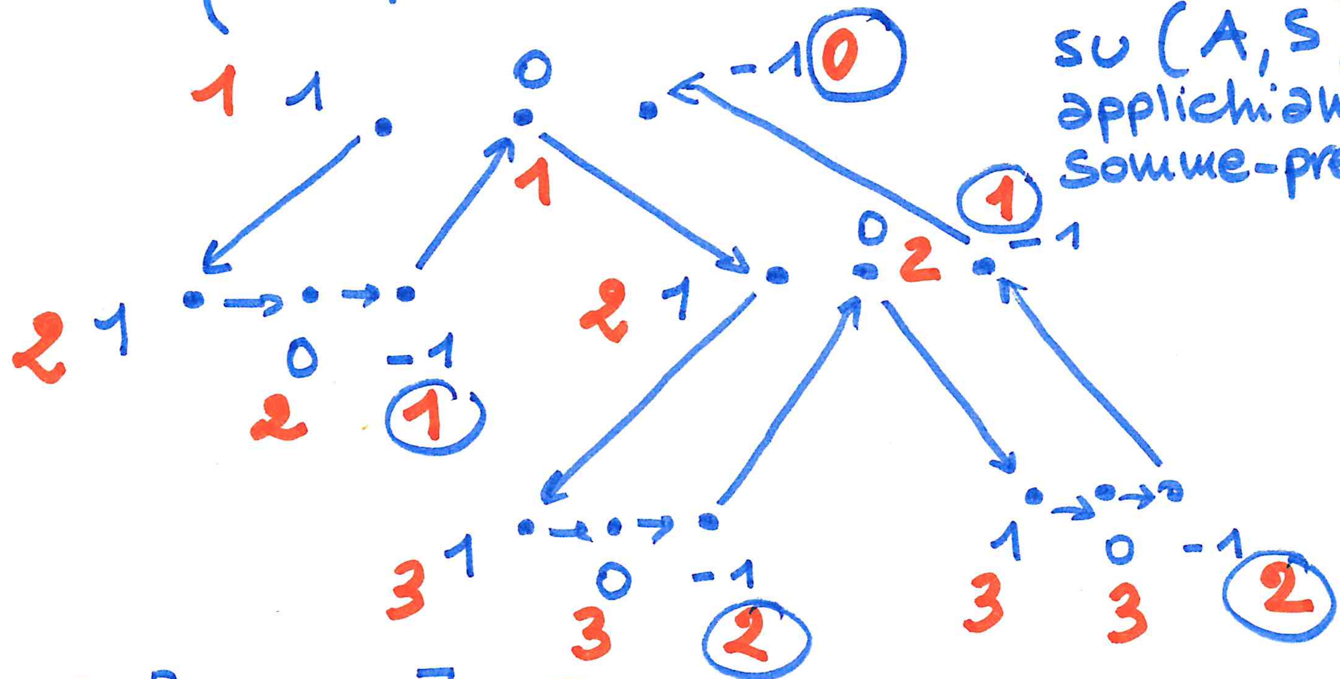
OTTIMALE

CALCOLO DELLA PROFONDITÀ DEI NODI

Definiamo un particolare array A:

$$A[(v,x)] = \begin{cases} 1 & \text{se } x = s \\ 0 & \text{se } x = c \\ -1 & \text{se } x = d \end{cases}$$





su (A, s) applichiamo somme-prefisse

$$A[(v,d)] = P(v)$$

Algoritmo parallelo per la profondità $P(v)$

1º calcolo A e S' (successore)

2º calcolo somme-prefisse su (A, S)

OUTPUT: è in $A[(v,s)]$ o $A[(v,d)]$

Prestazioni:

come prima:

$$E = \frac{n}{n/\log n \cdot \log n} \rightarrow C \neq 0$$

OTTIMALE

OSSERVAZIONI FINALI SU PRAM

1. Interesse Teorico :

   - processori sono uguali e alla pari
   - il tempo è strettamente legato alla computazione (comunicazione costante)

2. Interesse pratico :

   - realizzazione fisica dei multicore

La realizzazione dei multicore ha portato
l'interesse da ambiti scientifici ben precisi
(fisici, economici, militari, sociali)

ad un ambiente più ampio :
il semplice consumatore / informatico

In realtà :

Prima del 2000 :
si aumentava il clock con problemi :
- di assorbimento di energia (> 100 W)
- di raffreddolamento

Dopo il 2000:

si è aumentato il grado di parallelismo (multicore)

- clock con (volendo) minor frequenza
- minor assorbimento di energia
- vantaggi sul raffreddamento

$\Downarrow$

nuovi sviluppi teorici in ambito di algoritmi paralleli

- scrittura / riscrittura / manipolazione di software per i multicore

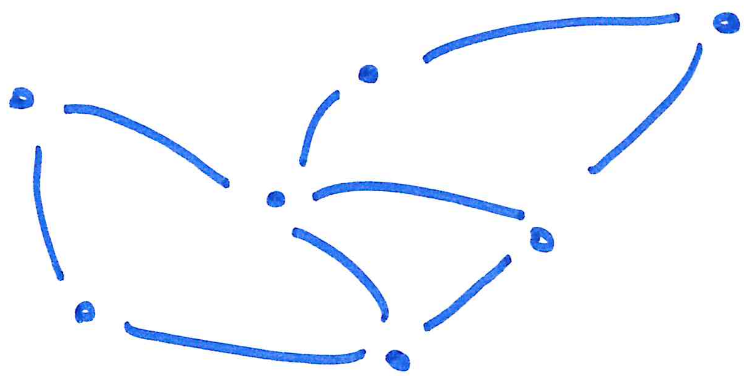
Architetture parallele a memoria distribuita

hardware dei supercomputer

anni 60': CRAY, INTEL PARAGON

Attuali: CRAY, BLUE GENE, RED STORM,
EARTH SIMULATOR, TIANHE-2, ...

Rappresentata da un grafo G



← nodi = processori

archi = rete di connessione

CARATTERISTICHE:

Processori: RAM sequenziali con:

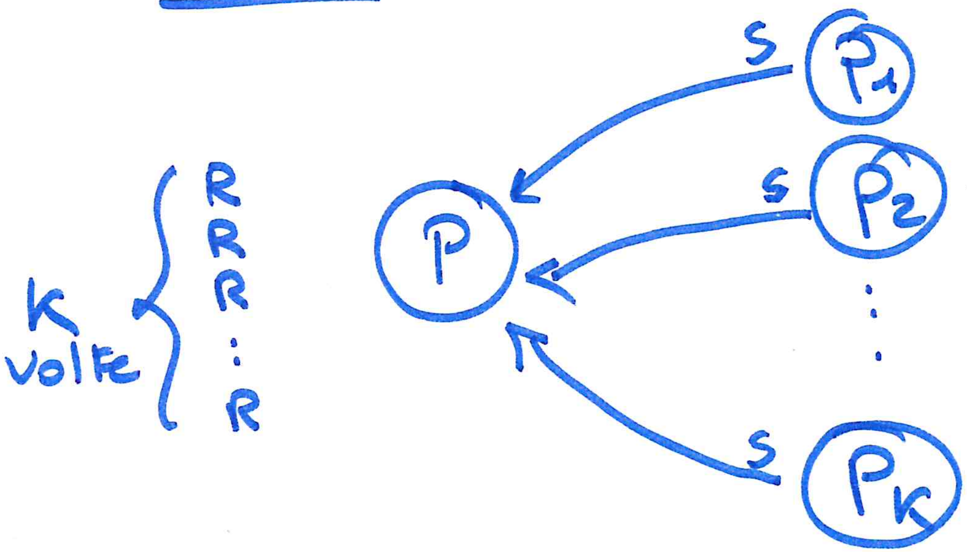
elementi di calcolo ← istruzioni per il calcolo + memoria privata

\+

e router ← istruzioni per la comunicazione:

send / receive

NOTA: anche la comunicazione avviene in parallelo



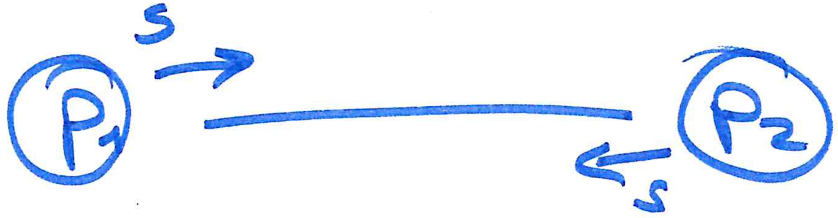
ma se $P_1, \ldots, P_K$ mandano contemp. dati a P con send simultanee

P lavorando sequenzialm. richiede K receive

⇓

K+1 passi per la comunic.

- collegamenti: di tipo full-duplex



se c'è un collegamento diretto, la comunicazione costa: 2 passi

	P_1	P_2
1° passo	S	S
2° passo	R	R

- clock centrale: scandisce il tempo per tutti i processori

- programma: come nelle PRAM:

for $k \in I$ par do
istruzione k ← SIMD

+ S e R

- CAMBIA INPUT / OUTPUT (manca la memoria condivisa)

INPUT: distribuito tra i processori

OUTPUT: o abbiamo un processore dedicato
o si legge in un certo ordine tra i vari processori

- Risorse di calcolo:
  - numero processori → può essere la lunghezza dell'input ma ci sono tecniche per abbassare il numero
  - Tempo dato da:

Tempo di calcolo
+ Tempo di comunicazione

può essere rilevante

è legato alla rete di connessione

Si definiscono i seguenti PARAMETRI DI RETE

Dato l'architettura $G = (V, E)$ definiamo:

- GRADO di G:

$$\gamma = \max \{ \rho(v) \mid v \in V \}$$

dove $\rho(v)$ = numero di archi incidenti su v

γ alto permette buone comunicazioni ma rende + difficile la realizzazione fisica

- DIAMETRO di G:

$$\delta = \max \{ d(v, w) \mid v \neq w \in V \}$$

$d(v,w)$ ↓ la più corta distanza tra v e w

valori bassi di δ sono da preferire
ma aumentano il parametro γ

- ampiezza di bisezione di G :

β = minimo numero di archi in G
che tolti mi dividono i nodi
in circa due metà

rappresenta la capacità di trasferire le inform.
in G, ancora una volta β alto preferibile ma
incrementa γ

TIPICI PROBLEMI che mettono in risalto pregi e difetti di queste architetture

- MAX : è richiesta una comunicazione a coppie di processori $\rightsquigarrow \delta$

- ORDINAMENTO : si richiede lo spostamento di parti dell'input $\rightsquigarrow \beta$

Valgono i seguenti limiti inferiori per i tempi di soluzione di questi due problemi.

FATTO I: Il tempo richiesto per risolvere MAX in G è almeno "δ"

dim: ogni coppia di processori deve comunicare

FATTO II: Il tempo richiesto per risolvere ORDINAMENTO in G è almeno $\frac{n}{2} \cdot \frac{1}{\beta}$

dim: