Parallelismo: Lema de Brent y Trabajo.

Note Title 11/11/2010

Recuerdo Def. TA, T^*, S, E.

Algoritmo 1 Max (n/2 procesadores)

- dividir el arreglo en $n/2$ pares
- reducir los candidatos a $n/2$
- iterar con $n/4$ procesadores.

$T^*(n) = n-1$

$T(n,n) = \lg n$

$S(n,n) = \frac{n-1}{\lg n}$

$E(n,n) = \frac{n-1}{n \lg n}$

Se puede mejorar?

- Si tenemos $p << n$ procesadores
  $\longrightarrow$ Lemma de Brent en 15mins
- En EREW, no se puede mejorar el tiempo (incluido con $O(n^2)$ procs)

$n/2$

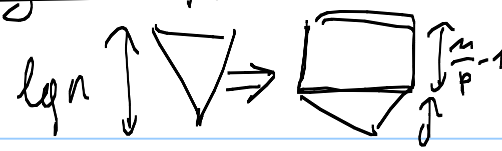
$n/4$

$n/8$

$\vdots$

(en total, mucho menos que $n \lg n$)

Algoritmo parallelo con mejor efficiencia



1. Dividir los n elementos en grupos de tamaño $\frac{n}{p}$ $O(1)$
2. Cada procesador calcula el max de su grupo (en tiempo $\frac{n}{p} - 1$)
3. Calcula el max de los p maximos usando los p procesadores con el algoritmo previo (en tiempo $\lg p$)

$$T(n,p) = \frac{n}{p} - 1 + \lg p \leq 2\lg n \qquad T^*(n) = n-1$$

Por $p = \dfrac{n}{\lg n}$

$$S(n,p) = \frac{n-1}{\frac{n}{p} - 1 + \lg p} = \frac{p(n-1)}{n - p + p\lg p} = \frac{n-1}{2\lg n}$$

$$E(n,p) = \frac{n-1}{n - p + p\lg p} = \frac{n-1}{2\lg n \cdot \frac{n}{\lg n}} = \frac{n-1}{2n} \approx \frac{1}{2}$$

<u>Tiempo</u> $= 2\lg n - \lg\lg n \in \Theta(\lg n)$

<u>Este Algoritmo es</u> <u>EREW</u>

② Lemma de Brent

El principio de "simular"

un grupo de procesadores con uno

se aplica en muchas aplicaciones.

$\Rightarrow$ $\left.\begin{matrix} T(n) \\ W(n) \end{matrix}\right\}$ son suficientes

(3) Ejemplo: Ranking en lista

Input: Lista enlazada.

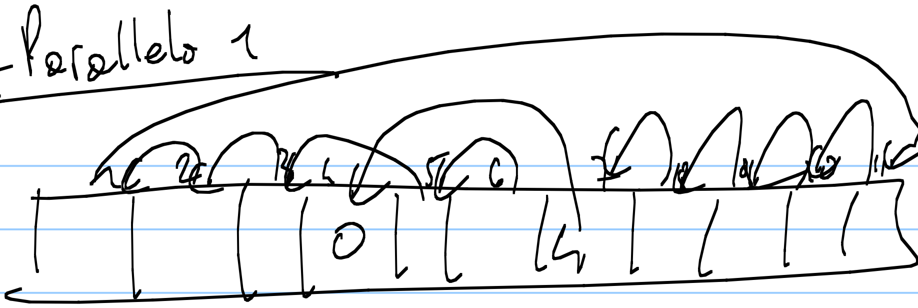
Output: el "rango" de cada elemento
= la distancia hasta el fin de la lista

Secuencialmente: n operaciones.

En Paralelo: suponga $A[1..n]$ con los valores
$N[1..n]$ con los punteros
Calculamos $R[1..n]$

a) Algoritmo Parallelo 1

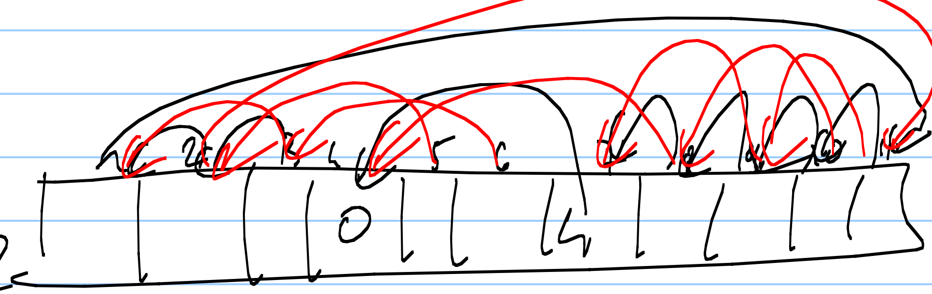
N_1



R_1

0	0	0	1	0	0	2	0	0	0	0

N_2



R_2

0	0	0	1	0	0	2	3	4	0	0

Cada procesador calcula un camino "corto" hasta el ultimo elemento de la lista, de tamaño a el maximo $\lg n$

$\longrightarrow T(n) = \log n$