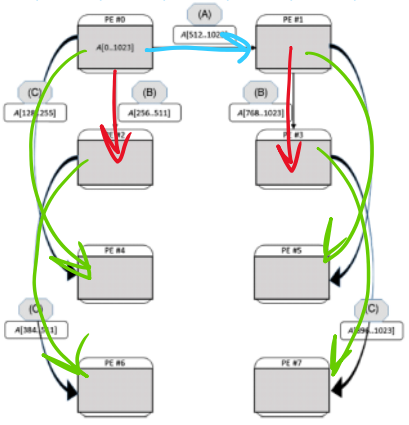
1.- Problema de 1.1

α: tiempo computo ; β: tiempo comunicar

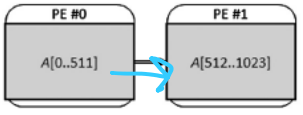
1. Analyze the speedup and the efficiency of the parallel summation algorithm presented in Section 1.1 using $n = 2048$ numbers assuming that each PE can add two numbers in one millisecond and each PE can send m numbers in $2 + m/1024$ milliseconds to another PE. Vary the number of PEs from 1 to 1024 using powers of 2.



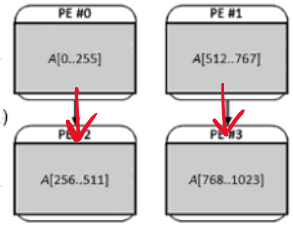
Pasos del algoritmo

0) nº de PE = p ; nº de datos = N

1) dividir los datos a la mitad: $\beta(\frac{N}{2})$



2) Repetir entre p hasta quedar con $\frac{N}{p}$ datos en cada PE



$\log_2 P$ es el nº de divisiones necesarias.

y cada vez habrían menos datos que transmitir

$$\beta\left(\frac{N}{2}\right) + \beta\left(\frac{N}{4}\right) + \beta\left(\frac{N}{8}\right) + \dots \beta\left(\frac{N}{p}\right)$$

$$T_2 = \sum_{i=1}^{\log_2 P} \beta\left(\frac{N}{2^i}\right) \quad \text{con } \beta(m) = 2 + \frac{m}{2^{10}}$$

$$T_2 = 2\log P + \sum^{\log_2 P} \frac{1}{2^{10}} \cdot \frac{N}{2^i} = 2\log P + N\left(\frac{1}{2}\right)^{10} \sum^{\log_2 P} \left(\frac{1}{2}\right)^i$$

$$T_2 = 2\log P + \sum_{i=1}^{\log_2 P} \frac{1}{2^{10}} \cdot \frac{N}{2^i} = 2\log P + N\left(\frac{1}{2}\right)^{10} \sum_{i=1}^{\log_2 P} \left(\frac{1}{2}\right)^i$$

donde $\sum_{i=1}^{N} a^i = a\,\frac{a^N - 1}{a-1}$

$$\sum_{i=1}^{\log_2 P} \left(\frac{1}{2}\right)^i = \frac{1}{2}\,\frac{\left(\frac{1}{2}\right)^{\log_2 P} - 1}{\frac{1}{2} - 1} = \frac{1}{2}\,\frac{2^{-\log P} - 1}{-\frac{1}{2}} = 1 - 2^{-\log P}$$

$$= 1 - \frac{1}{P} = \frac{P-1}{P}$$

$$\therefore \boxed{T_2 = 2\log P + N\left(\frac{1}{2}\right)^{10}\left(\frac{P-1}{P}\right)}$$

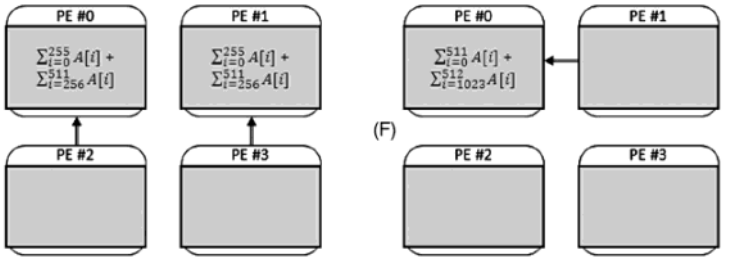
tiempo en [ms] en repartir los N datos

// $\log P = \log_2 P$

3) Sumar los $\frac{N}{P}$ nº dentro de PE:

$$\boxed{T_3 = \left(\frac{N}{P} - 1\right)\alpha = \frac{N}{P} - 1 \text{ [ms]}} \;//$$

4) Recuperar y resumar.



4a) Transmitir 1 nº

4b) Sumar 2 nº

⋮

repetir $\log_2 P$ veces.

$\log_2 P$ veces.

$$T_4 = \log P \cdot \left(\beta(1) + \alpha\right)$$

$$T_4 = \log P \left(2 + \frac{1}{2^{10}} + 1\right) \Rightarrow \boxed{T_4 = \left(3 + \frac{1}{2^{10}}\right) \log P \quad [ms]}$$

así el tiempo total:

$$T_p = 2 \log P + N \left(\frac{1}{2}\right)^{10} \left(\frac{P-1}{P}\right) + \frac{N}{P} - 1 \quad + \left(3 + \frac{1}{2^{10}}\right) \log P$$

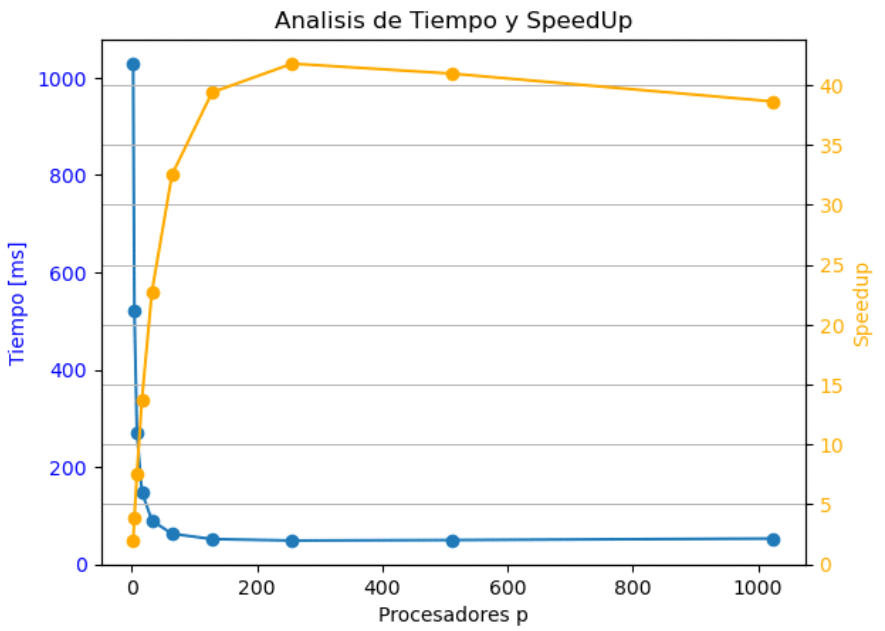
$$T_p = \left(5 + \frac{1}{2^{10}}\right) \log P + \frac{N}{P} \left((P-1)\left(\frac{1}{2}\right)^{10} + 1 \right) - 1 \quad / \cdot [ms]$$

Comparado al algoritmo en 1 solo PE

$$T_1 = (N-1)\alpha = N-1 \quad / \cdot [ms]$$

el Speed up:

$$S = \frac{T(1)}{T(P)} = \frac{N-1}{\left(5 + \frac{1}{2^{10}}\right) \log P + \frac{N}{P} \left((P-1)\left(\frac{1}{2}\right)^{10} + 1 \right) - 1}$$



la eficiencia

$$E = \frac{S}{P}$$