Programación Paralela (2015-2016)
Lenguajes y Sistemas de Información

Grado en Ingeniería Informática

E. T. S. de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Trabajo propuesto: Cómo afecta el tamaño del problema y el número de procesos a la eficiencia de un algoritmo paralelo

Francisco Javier Bolívar Lupiáñez

9 de marzo de 2016

Índice

Índice de figuras

1. Planteamiento del problema

Se pretende demostrar por qué la eficiencia (E) tiende a aumentar al aumentar el tamaño del problema (N) y a disminuir al aumentar el número de procesos (P).

2. Demostración

2.1. Modelo matemático

Para demostrarlo se va a plantear un ejemplo y se utilizará el del **algoritmo del patrón de once puntos** visto en clase con una pequeña modificación: **la matriz tiene el mismo tamaño en todas las direcciones** (NxNxN). Este algoritmo realiza operaciones sobre una matriz 3D y para computar cada punto hace falta conocer el valor de sus vecinos de arriba y abajo, dos a la izquierda, dos a la derecha, dos hacia delante y dos hacia atrás, además del propio punto (Figura 2.1).

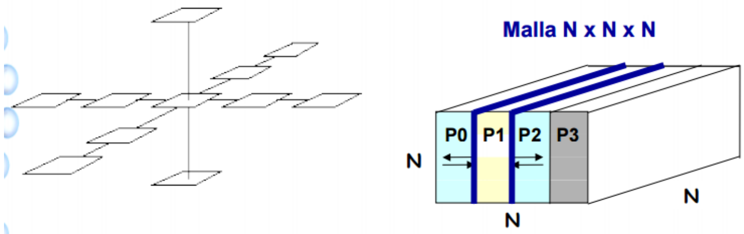

Figura 2.1: Patrón de once puntos, tamaño de la malla y reparto de ésta entre los procesos

El tiempo de un algoritmo secuencial sería (T_1):

$$T_1 = t_c N^3 \tag{2.1}$$

Cuando se paraleliza, se reparten bloques de la matriz como se muestra en la Figura 2.1, por tanto cada proceso necesitará intercambiar $2N^2$ puntos con dos vecinos. Lo que hace que el tiempo de comunicación (T_{comm}) sea:

$$T_{comm} = 2(t_s + t_w 2N^2) \tag{2.2}$$

Y el tiempo de computación (T_{comp}):

$$T_{comp} = \frac{t_c N^3}{P} \tag{2.3}$$

Por tanto el tiempo del algoritmo paralelizado (T_p) sería:

$$T_p = T_{comp} + T_{comm} = \frac{t_c N^3}{P} + 2t_s + t_w 4N^2 \tag{2.4}$$

La ganancia de velocidad (S) se calcula como la división del algoritmo secuencial (T_1) entre el paralelo (T_p):

$$S = \frac{T_1}{T_p} = \frac{t_c N^3}{\frac{t_c N^3}{P} + 2t_s + t_w 4N^2} \tag{2.5}$$

Y la eficiencia (E), finalmente, se calcularía como la división entre la ganancia de velocidad (S) entre el número de procesos (P):

$$E = \frac{S}{P} = \frac{t_c N^3}{P(\frac{t_c N^3}{P} + 2t_s + t_w 4N^2)} = \frac{N^3 t_c}{N^3 t_c + 2Pt_s + 4PN^2 t_w} \tag{2.6}$$

En esta función resultante contamos con cinco variables:

- t_c: tiempo de computación por punto.
- t_s: tiempo de inicialización de comunicación.
- t_w: tiempo de comunicación por palabra.
- N: tamaño de la matriz.
- P: número de procesos.

Las tres primeras no varían, y se va a suponer que $t_c = 1$ y $t_s = t_w = 0.5$. Por lo que vamos a utilizar la siguiente función para obtener los tiempos variando N y P:

$$f(N, P) = \frac{N^3}{N^3 + P + 2PN^2} \tag{2.7}$$

2.2. Mediciones y resultados

Se toman tiempos con 10, 40, 70 y 100 procesos y con tamaños de 100, 400, 700 y 1000. Obteniendo los siguientes resultados:

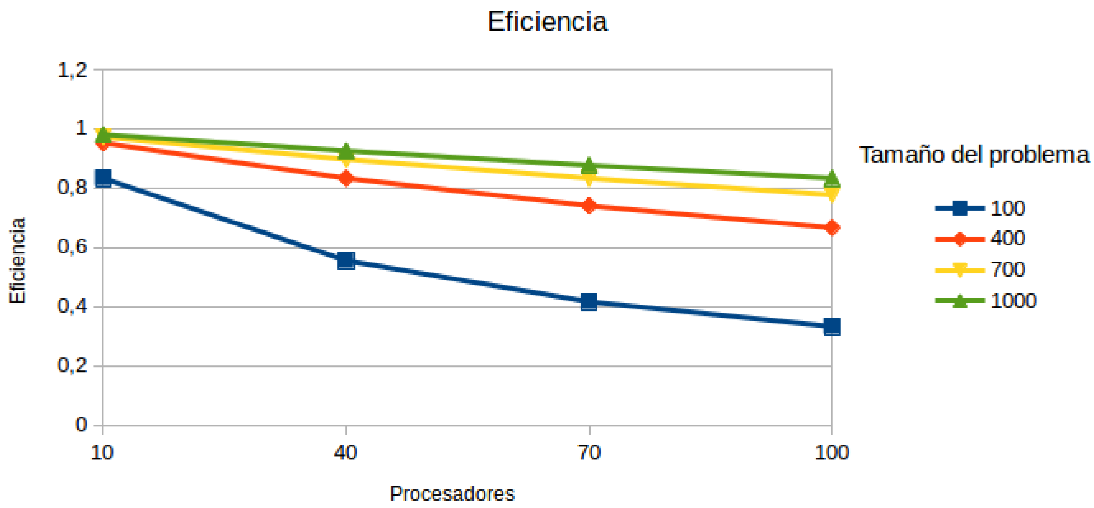

Figura 2.2: La eficiencia disminuye conforme se aumenta el número de procesos

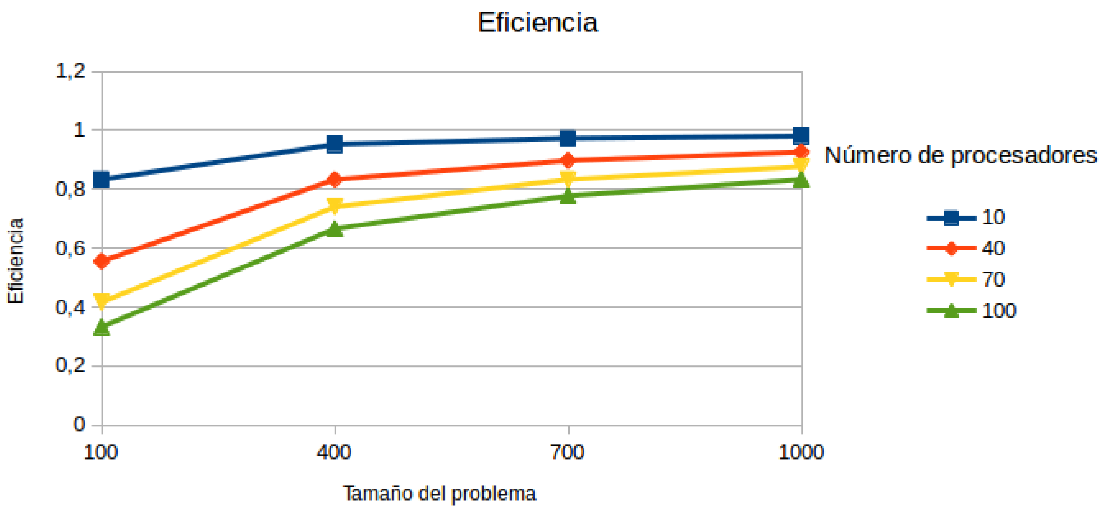

Figura 2.3: La eficiencia aumenta conforme se aumenta el tamaño del problema

3. Conclusiones

La eficiencia se define como la **fracción de tiempo en el que los procesos realizan un trabajo útil**.

Por ello, es obvio pensar que a más procesos, más tiempo de comunicación y, por tanto, menos tiempo de trabajo útil. Los resultados obtenidos son coherentes respecto a esta afirmación y podemos ver como para todos los tamaños, la eficiencia disminuye.

Al mismo tiempo, se puede presuponer que, de acuerdo a la definición que se da de eficiencia, al aumentar el tamaño del problema más carga de trabajo útil tendrán los procesos y por tanto mayor será el porcentaje de su tiempo dedicado a ello.