

# Analisis de Tiempos en algoritmos usando OpenMP en C

Alfred A. Chavez <sup>1</sup>

<sup>1</sup>Ciencia de la computación – Universidad Católica San Pablo

alfred.chavez@ucsp.edu.pe

## 1. Introducción

En este documento se presentarán los resultados de las pruebas hechas en algoritmos seriales, técnicas y algoritmos en paralelo, usando OpenMP

## 2. Pruebas

Las siguientes pruebas son de códigos hechos en C++, en una laptop con procesador Intel Core i5-7200U, 2.50GHz x 4(cores), RAM 11.6GB, con sistema operativo Linux 4.15.3, distro Fedora 28; comopilador usado es GNU GCC 7.3.1 y para la medida de tiempo se usa la herramienta brindada por OpenMP:

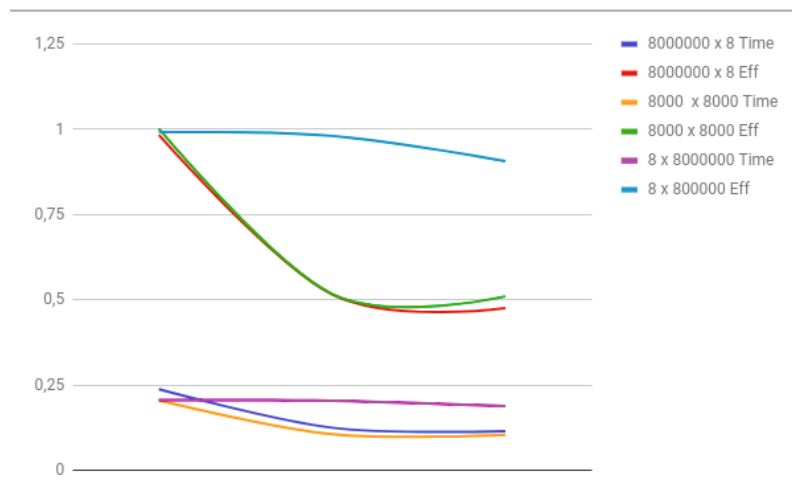
```
1 | gcc -o a.out -fopenmp -lpthread -g
```

### 2.1. Resultados

#### 2.1.1. Tiempo

| Threads | Matrix Dimension |              |             |              |             |              |
|---------|------------------|--------------|-------------|--------------|-------------|--------------|
|         | 8000000 x 8      |              | 8000 x 8000 |              | 8 x 8000000 |              |
|         | Time             | Eff          | Time        | Eff          | Time        | Eff          |
| 1       | 0,2373607        | 0,9835045118 | 0,20450675  | 1,002081271  | 0,2057724   | 0,9917710042 |
| 2       | 0,1245578        | 0,5161054811 | 0,10550585  | 0,5169777344 | 0,203497    | 0,9808041508 |
| 4       | 0,1146799        | 0,4751763837 | 0,1040724   | 0,5099538421 | 0,18808285  | 0,9065118403 |

Figura 1. Tabla de tiempos matriz vector



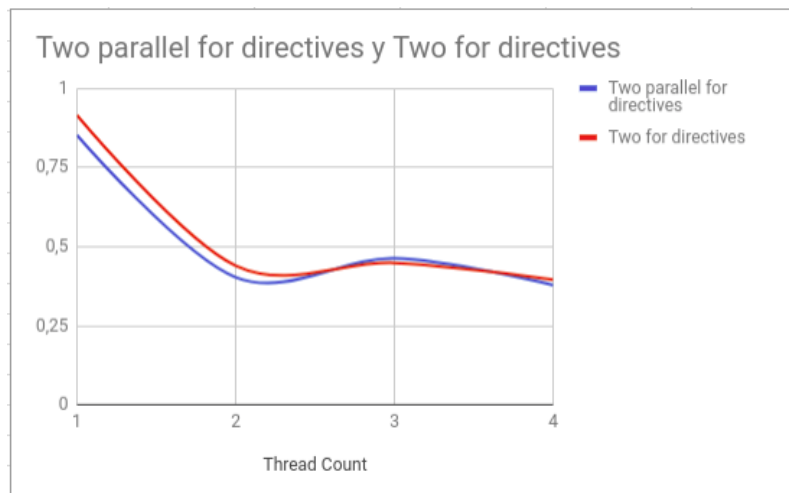
**Figura 2. Comparacion de tiempos Matrix Vector**

| Thread Count                | 1          | 2          | 3          | 4         |
|-----------------------------|------------|------------|------------|-----------|
| Two parallel for directives | 0,85251535 | 0,40369695 | 0,46362495 | 0,379155  |
| Two for directives          | 0,915412   | 0,4401302  | 0,448995   | 0,3961056 |

**Figura 3. Tabla de tiempos odd-even sort**

## 2.2. Análisis

Podemos ver en la multiplicación Matriz-vector, que los tiempos se reducen considerablemente para las matrices de tamaño 8000000 x 8 y 8000 x 8000 conforme aumenta el número de threads, y en la de 8 x 8000000 también se reduce el tiempo pero no de forma muy notoria como en las otras matrices, y con respecto a la eficiencia se presenta el mismo comportamiento; en el ordenamiento odd-even podemos ver que los tiempos en ambos algoritmos no se diferencian mucho unos de otros, estos disminuyen conforme aumenta el número de threads, de forma similar.



**Figura 4. Comparacion de tiempos odd-even sort**