Analisis de Tiempos en algoritmos en MPI C++

Alfred A. Chavez [1]

[1]Ciencia de la computación – Universidad Católica San Pablo

alfred.chavez@ucsp.edu.pe

1. Introducción

En este documento se presentará un análisis de performance, en tiempo en dos algoritmos, Multiplicación Matriz-Vector y ordenamiento Odd-Even.

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 27; comopilador usado es GNU GCC 7.3.1 con los flags que brinda mpich 3.2.1 (mpicxx):

```
mpicxx <source code>
```

Para las mediciones de tiempo se hizo uso de MPI_WTime.

2.1. Código

2.1.1. MPI Matriz x Vector y MPI Odd Even Transposition Sort

https://github.com/alfredchavez/paralelos

2.2. Resultados

2.2.1. Tiempo

Cuadro 1. Tiempos de ejecución Matriz Vector

Tamaño de vector = num. procesos	Tiempo
1	0,000033 ms
5	0,10300 ms
10	0,569339 ms
15	1,821512 ms
20	2,188994 ms

2.3. Conclusión

Vemos que en la multiplicación matrix vector el tiempo se estabiliza, va disminuyendo conforme va creciendo el tamaño del vector, asi como el número de procesos que se usa para correr el algortitmo.
También que en el algoritmo Odd-Even se puede ver el comportamiento de la función va creciendo conforme crece el tamaño del array.

Cuadro 2. Tiempos de ejecución Odd-Even

Tamaño de vector(doble num. procesos)	Tiempo
2	0,000042 ms
10	1,425324 ms
20	12,985810 ms
30	34,784391 ms
40	79,660144 ms

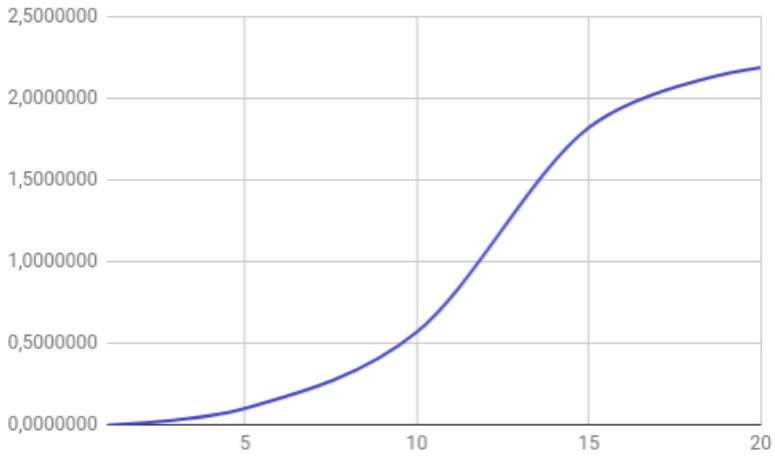

Figura 1. Comparacion de tiempos matriz vector

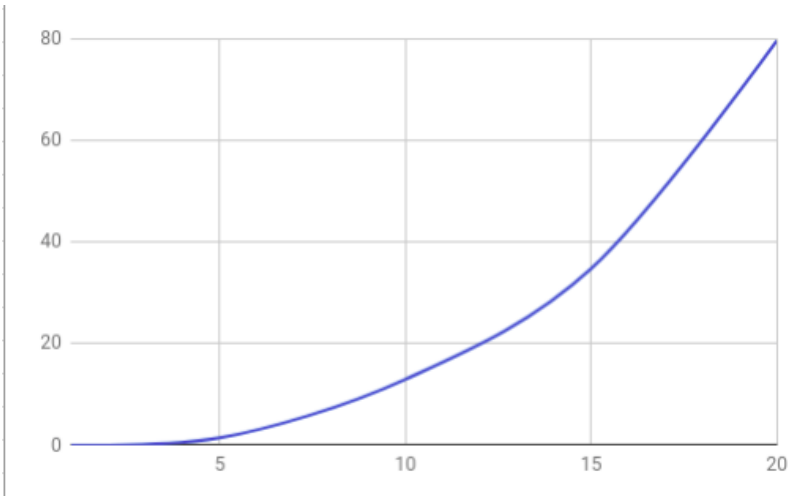

Figura 2. Comparacion de tiempos odd even