

# Cómputo Concurrente 2024-2

## Práctica 2

### Ley de Amdahl y Paralelismo

Natalia Abigail Pérez Romero  
Jonathan Bautista Parra  
Valeria Reyes Tapia

24 de febrero de 2024

## 1. Introduction

En esta práctica se implementó un programa en Java que simula un ataque de fuerza bruta para obtener una contraseña. El programa utilizará hilos para realizar el ataque de manera concurrente de forma que cada hilo se encarga de buscar una parte de las combinaciones posibles de la contraseña. También se calculará la aceleración teórica y obtenida en función del número de hilos y porcentaje de código en paralelo.

## 2. Graficas y tablas de valores

Después de ejecutar la solución para 1, 2, 27 y 100 hilos, se obtuvieron los siguientes resultados:

Número de hilos	Tiempo de ejecución (s)
1	5.79159E+11
2	7.2579E+11
27	3.64659E+11
100	1.21143E+12

Cuadro 1: Tabla de tiempos de ejecución en función del número de hilos.

Nuestra contraseña tiene 6 caracteres, por lo que el número de combinaciones posibles es  $26^6 = 308915776$ . En la tabla 2 se muestra la aceleración teórica y obtenida en función del número de hilos y porcentaje de código en paralelo. En este caso, consideramos el porcentaje de código paralelizable como el 100 %, dado que los hilos no requieren de comunicación entre ellos para realizar su trabajo.

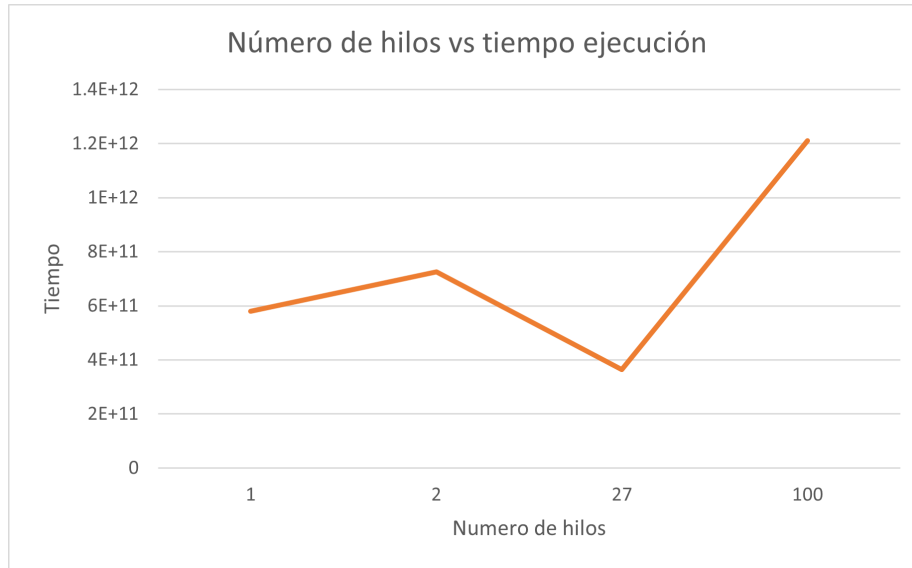


Figura 1: Gráfica de tiempos de ejecución en función del número de hilos.

Número de hilos	Aceleración teórica	Aceleración obtenida	% código en paralelo
1	1	5.79159E+11	100 %
2	$(5.79159E+11/2) = 2.8958E+11$	7.2579E+11	100 %
27	$(5.79159E+11/26) = 2.14133E+10$	3.64659E+11	100 %
100	$(5.79159E+11/100) = 2.14133E+10$	1.21143E+12	100 %

Cuadro 2: Tabla de aceleración teórica y obtenida en función del número de hilos y porcentaje de código en paralelo.

Si el tiempo de ejecución con un hilo es de 5.79159E+11, entonces la aceleración teorica con 2 hilos es de 2.89579E+11, con 27 hilos es de 2.14133E+10 y con 100 hilos es de 5.79159E+9.

### 3. Cuestionario

1. ¿Cuál fue mi contraseña?

La contraseña es: hlsbrr

2. ¿Cuántas posibles contraseñas hay?

Si la contraseña tiene entre 7 y 13 caracteres, por lo que el numero de combinaciones posibles es:

$$\sum_{i=7}^{13} 26^i = 2147583647$$

Nuestra contraseña es de 6 caracteres, por lo que el numero de combinaciones posibles es:

$$26^6 = 308915776$$

3. ¿La ley de Amdahl siempre se cumple?

No. La ley de Amdahl no siempre se cumple en la práctica. La ley establece que la mejora de la velocidad de un componente de un sistema solo puede mejorar el rendimiento general del sistema hasta un límite determinado. El límite está determinado por la fracción del tiempo que se ejecuta el componente en cuestión.

4. ¿En qué casos no se cumple?

La introducción de paralelismo puede llevar la necesidad de sincronización entre hilos o procesos, la comunicación entre ellos y la gestión de recursos compartidos. Esto puede contrarrestar parcial o totalmente los beneficios del paralelismo. Problemas con dependencias de datos, donde la salida de una operación es necesaria como entrada para otra, pueden limitar la capacidad de paralelización.

5. ¿Por qué crees a que se debe esto?

Los sistemas no son perfectamente paralelos: La mayoría de los sistemas tienen algunos componentes que son secuenciales. Los componentes no siempre se ejecutan al máximo rendimiento: Los componentes pueden estar inactivos o ejecutarse a una velocidad inferior a su máximo rendimiento.

6. ¿Cuál sería la mejora máxima? Es decir, la aceleración teórica máxima

La mejora máxima o aceleración teórica máxima según la Ley de Amdahl se puede calcular utilizando la fórmula proporcionada por la misma ley:

$$S = \frac{1}{F_s - \frac{F_p}{P}}$$

Donde:

- S es la mejora máxima
- $F_s$  es la fracción del código que se ejecuta de manera secuencial.
- $F_p$  es la fracción del código que se puede paralelizar.
- P es el número de procesadores

La mejora máxima se alcanza cuando P tiende a infinito, es decir, cuando se utiliza un número infinito de procesadores:

$$S = \frac{1}{F_s}$$

Este resultado muestra que la mejora máxima está inversamente proporcional a la fracción secuencial del código ( $F_s$ ). Cuanto menor sea ( $F_s$ ), mayor será la mejora máxima posible.

7. Escribe tus conclusiones, además de lo que aprendiste en esta práctica, contratiempos y descubrimientos que hubo durante su realización.

En esta práctica aprendimos a utilizar hilos en Java y a calcular la aceleración teórica y obtenida en función del número de hilos y porcentaje de código en paralelo. Aprendimos que la ley de Amdahl no siempre se cumple en la práctica y que la mejora máxima o aceleración teórica máxima se alcanza cuando se utiliza un número infinito de procesadores.

Durante la realización de la práctica, tuvimos problemas con la implementación de los hilos porque nuestro poder de computo es limitado, así que nuestra contraseña tuvo que ser de 6 caracteres, lo que limitó el número de combinaciones posibles.

8. ¿Cuál es su rol?

Nuestro rol es el de los hilos, los cuales se les encarga el trabajo de obtener por fuerza bruta la contraseña para obtener un mensaje secreto.

9. ¿Cuál es mi rol?

Tu rol es el del despachador, el cual se encarga de repartir el trabajo entre los hilos.

### 3.1. Una captura de pantalla cuando agregaste tus palabras.

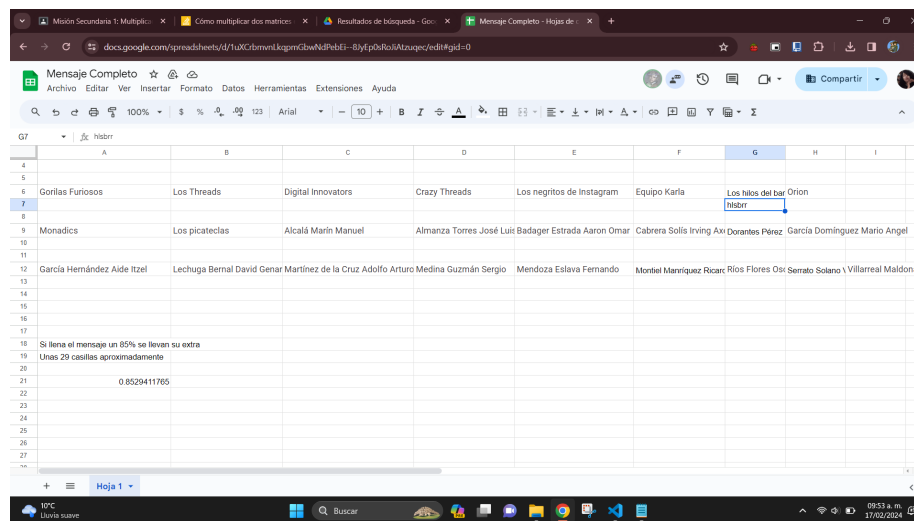


Figura 2: Captura de pantalla de cuando se agregaron las palabras.

Corrección:

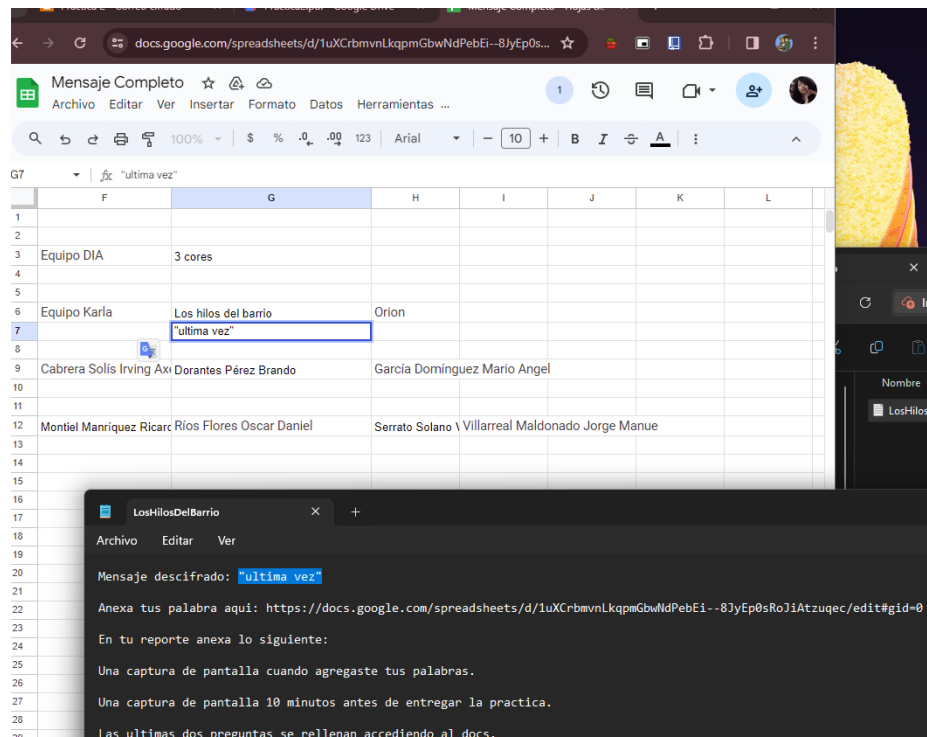


Figura 3: Captura de pantalla de cuando se agregaron las palabras.

### 3.2. Una captura de pantalla 10 minutos antes de entregar la practica.

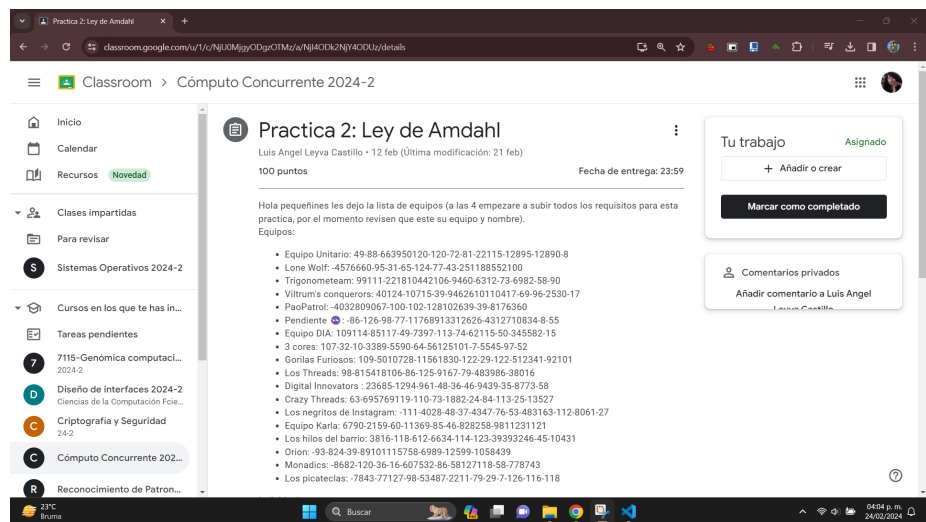


Figura 4: Captura de pantalla 10 minutos antes de entregar la práctica.