**Caso de estudio multiplicación de Matrices: forks vs implementación secuencial**

#### Oscar M. Giraldo Herrera

Universidad Tecnológica De Pereira

Facultad De Ingenierías

Andrés Ramiro Barrios

#### Octubre 2021

# TABLA DE CONTENIDO

Contenido

[TABLA DE CONTENIDO 2](#_Toc89436174)

[RESUMEN 4](#_Toc89436175)

[ABSTRACK 4](#_Toc89436176)

[INTRODUCCIÓN 5](#_Toc89436177)

[MARCO CONCEPTUAL 7](#_Toc89436178)

[¿QUE ES SOFTWARE? 7](#_Toc89436179)

[¿QUE ES LA SUPERCOMPUTACION? 7](#_Toc89436180)

[¿QUE ES LA COMPUTACION SECUENCIAL? 8](#_Toc89436181)

[COMPUTACION SECUENCIAL 9](#_Toc89436182)

[COMPUTACION PARALELA 11](#_Toc89436183)

[¿QUE ES PARALELISMO? 11](#_Toc89436184)

[¿QUÉ ES LA COMPUTACION PARALELA? 12](#_Toc89436185)

[¿COMO SE REALIZA LA MULTIPLICACION DE DOS MATRICES? 13](#_Toc89436186)

[¿QUÉ ES LA LEY DE AMDAHL? 14](#_Toc89436187)

[¿QUÉ ES HIGH PERFORMANCE COMPUTING? 16](#_Toc89436188)

[¿QUÉ SON LOS HILOS Y LA CONCURRENCIA DE MULTIPLES HILOS? 16](#_Toc89436189)

[MARCO CONTEXTUAL 18](#_Toc89436190)

[DESARROLLO 21](#_Toc89436191)

[PRUEBAS 21](#_Toc89436192)

# RESUMEN

El presente documento tiene como objetivo mostrar el proceso de desarrollo, implementación y ejecución de un programa que realice la multiplicación de matrices de tamaño N x N, en forma secuencial y paralela, y la recolección de datos del tiempo de ejecución en CPU del programa, para hallar la tasa de aceleración (speed up rate) propuesto en la ley de amdahl, como caso de estudio para la materia high performance computing (HPC)

PALABRAS CLAVE: high performance computing, desarrollo, programación en C, Python, computación secuencial, computación paralela, threads, hilos, sistema computacional, CPU, lenguaje de programación, paralelismo, concurrencia, hardware, software, speed up rate.

# ABSTRACK

*The purpose of this document is to show the process of development, implementation and execution of a program that performs the multiplication of matrices of size N x N, sequentially and in parallel, and the collection of data on the CPU execution time of the program, to find the acceleration rate (speed up rate) proposed in Amdahl's law, as a case study for the subject of high performance computing (HPC).*

*KEY WORDS: high performance computing, development, C programming, python, sequential computing, parallel computing, threads, computational system, CPU, programming language, parallelism, concurrency, hardware, software,* speed up rate

# INTRODUCCIÓN

James Nicholas Graymás conocido como **Jim Gray**, fue un científico de la computación estadounidense Recibió el Premio Turing en 1998 "por contribuciones originales a la investigación en bases de datos y procesamiento de transacciones, y su liderazgo técnico en la implementación de sistemas".

Gray estudió en la Universidad de California, Berkeley, donde recibió su licenciatura en ingeniería matemática en 1966 y su doctorado en ciencias de la computación en 1969. Fue el primer doctorando del departamento de ciencias de la computación en Berkeley. Ref. https://es.wikipedia.org/wiki/Jim\_Gray

Uno de los aportes a la ciencia de la computación que dejo Gray antes de su desaparición en el año 2007 fueron “los cuatro paradigmas de la ciencia” los

Que cuales son una recapitulación sobre la evolución que ha tenido la ciencia, desde sus comienzos de forma empírica, pasando por la rama de la teoría y llegando hasta la rama computacional, esta última permitiéndonos avanzar hacia lo que Gray llamo la exploración de datos o e- science.

Gracias a la existencia de estos sistemas computacionales, podemos realizar simulaciones de sistemas más complejos, que de forma empírica tardaríamos años en conocer resultados, evaluarlos, compararlos, y lograr entender el funcionamiento de aquello sobre lo que queremos saber un poco más.

En matemática, una matriz es un arreglo bidimensional de números. Dado que puede definirse tanto la suma como el producto de matrices, en mayor generalidad se dice que son elementos de un anillo, Es larga la historia del uso de las matrices para resolver ecuaciones lineales. Un importante texto matemático chino que proviene del año 300 a. C. a 200 a. C., *Nueve capítulos sobre el Arte de las matemáticas* (*Jiu Zhang Suan Shu*), es el primer ejemplo conocido de uso del método de matrices para resolver un [sistema de ecuaciones simultáneas](https://es.wikipedia.org/wiki/Sistema_lineal_de_ecuaciones). Ref <https://es.wikipedia.org/wiki/Matriz_(matem%C3%A1ticas>). En promedio un ser humano que haya estudiado matrices y su aplicación, puede tardar entre 1 y 2.5 minutos en resolver la multiplicación de dos matrices pequeñas de un tamaño 2x2 o 3x3 con datos enteros de no más de dos dígitos, sin utilizar calculadora. Ref <https://www.youtube.com/watch?v=4HgEfOukm78> . Si en nuestro ámbito científico quisiéramos investigar algún concepto que requiera la multiplicación de matrices de gran tamaño, por ejemplo, por encima de los mil datos, y además con números de hasta 4 y 5 dígitos, tardaríamos muchísimo tiempo en recopilar la información necesaria para nuestra investigación, es aquí donde los sistemas computacionales entran a resolver por nosotros los cálculos matemáticos, de forma precisa y rápida.

Pero ¿qué tan rápido podría un sistema computacional resolver nuestra multiplicación de matrices? ¿Acaso estamos utilizando toda la capacidad del sistema para resolverlo o solo una parte? ¿Es posible mejorar el tiempo que tarda nuestro sistema en resolver la multiplicación?

Este documento mostrara los resultados obtenidos al desarrollar la multiplicación de matrices para cantidades de datos diferentes, aplicada primero de forma secuencial y enfrentándola contra una versión paralela de la misma multiplicación en la que buscaremos disminuir los tiempos de respuesta del sistema y aplicar los conceptos de speed up rate revisados en la ley de Amdahl, la cual propone que dependiendo del tamaño o porción de código que se logre paralelizar se obtendrá cierta aceleración en tiempos de ejecución y que llegando a un 95% de código paralelizado se podría alcanzar una aceleración de 20x de speed up rate.

# MARCO CONCEPTUAL

A continuación, se definen los términos y conceptos fundamentales para esclarecer y precisar las expresiones, ocupaciones y alcances de las herramientas usadas para el desarrollo de los programas, su ejecución y la recolección de datos de tiempo de ejecución de CPU tanto de la multiplicación secuencial como de la ejecución paralela.

## ¿QUE ES SOFTWARE?

Es un conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora (Real Academia Española). El software, El Alma del Computador. Así como la esencia del ser humano no está en su cuerpo sino en un algo abstracto llamado alma, es en un elemento intangible llamado software donde radica la mayor parte de la magia que ha convertido al computador en la herramienta más poderosa de nuestro tiempo. (EL TIEMPO, 1996).

## ¿QUE ES LA SUPERCOMPUTACION?

El término supercomputador se refiere al ordenador perteneciente a la gama más alta en cuanto a rendimiento, capacidad de almacenamiento y eficiencia posibles, de todos aquellos disponibles en un momento determinado del tiempo. Por lo tanto, el atributo super debe considerarse como una característica definitoria de un computador que dispone de la más alta tecnología existente en el mercado.

La arquitectura de los ordenadores secuenciales ha mejorado mucho desde las primitivas máquinas de los años 40 y 50. Existen, sin embargo, limitaciones físicas, tales como la velocidad de la luz, que constituyen una barrera natural para hacer del procesamiento secuencial algo ilimitado. El paralelismo ofrece hoy la forma de incrementar la potencia computacional más allá de esas barreras. Procesamiento paralelo es la solución de un único problema mediante el uso de más de un elemento de proceso y de técnicas de segmentación o de empleo repetido de procesadores. Conceptualmente el procesamiento paralelo es algo muy sencillo: subdivisión de un problema en un cierto número de subproblemas. Lo que resulta más complicado es la implementación de ese concepto en un sistema multiprocesador. Para resolver el problema es necesaria la cooperación entre los elementos de proceso o los procesadores. Con el paralelismo y la supercomputación es posible resolver, no solo los problemas que se nos presentan hoy día en fracciones del tiempo empleado hace algunos años, sino también otros más complejos e insolubles en el pasado.

Ref https://www.acta.es/medios/articulos/informatica\_y\_computacion/030009.pdf

REF https://www.youtube.com/watch?v=b5bQdTL0wAg

## ¿QUE ES LA COMPUTACION SECUENCIAL?

### COMPUTACION SECUENCIAL

En las ciencias de la computación, el acceso secuencial significa que un grupo de elementos es accedido en un predeterminado orden secuencial. Indica que el procesador que debe ejecutar de forma consecutiva una lista de acciones (Estas pueden ser a su vez otras estructuras de control). A veces es la única forma de acceder a los datos. Para construir una secuencia de acciones basta con escribir cada acción en una línea diferente.

Por último, es necesario señalar un aspecto importante de la composición secuencial y es  
que no es conmutativa.

En las estructuras de datos, se dice que una estructura tiene acceso secuencial si solo podemos visitar los valores contenidos en un determinado orden. El ejemplo trivial, es la lista enlazada.  
  
A pesar de que la computación secuencial ha alcanzado avances extraordinarios en nuestra historia reciente, no ha podido demostrar su aplicación exitosa en problemas complejos donde se requiera máquinas que funcionen de manera autónoma, robusta y eficiente, dentro de ambientes dinámicos y  
  
amenazantes. Existen otros paradigmas alternativos de cómputo, agrupados en lo que se ha denominado bioinformática y computación celular, los cuales tienen una motivación fundamentalmente biológica, y ciertamente han mostrado resultados prometedores. En este artículo se abordan sus principales características y se mencionan algunos avances.

Forma

Descripción generada automáticamente con confianza media

Ilustración 2 Esquema Secuencial

Gráfico

Descripción generada automáticamente

## COMPUTACION PARALELA

En el sentido más simple, la computación paralela es el uso simultaneo. Operando sobre el principio de problemas grandes, a menudo se pueden dividir en unos mas puequeños que luego son resueltos simultáneamente (en paralelo).

Un problema se divide en partes discretas que se pueden resolver simultáneamente.

* Cada parte se descompone en una serie de instrucciones
* Las instrucciones de cada parte se ejecutan simultáneamente en diferentes procesadores.
* Se emplea un mecanismo global de control/coordinación

### ¿QUE ES PARALELISMO?

El paralelismo es una forma de computación en la cual varios cálculos pueden realizarse simultáneamente,1 basado en el principio de dividir los problemas grandes para obtener varios problemas pequeños, que son posteriormente solucionados en paralelo. Hay varios tipos diferentes de paralelismo: nivel de bit, nivel de instrucción, de datos y de tarea. El paralelismo ha sido empleado durante muchos años, sobre todo para la Computación de alto rendimiento.

Diagrama

Descripción generada automáticamente

El paralelismo tiene res formas diferentes de aplicarse:

1. **Paralelismo independiente:** paralelismo-And se denomina independiente, debido a que su principal característica es que, una vez establecidas las dependencias entre variables, únicamente se permite ejecutar en un paralelo aquellas metas que sean independientes, de forma en que sus resultados no sean inconsistentes entre sí.
2. **Paralelismo Regular:** Se trata de aplicar paralelismo operaciones con datos donde sus procesos son mutuamente independientes.

Poniendo como ejemplo la condición para que dos rectas sean paralelas si sus vectores son paralelos, es decir si estos son linealmente independientes.

1. **Paralelismo no Estructurado:** Es cuando las computaciones concurrentes difieren, significando que el acceso a los datos no es predecible y necesitan ser coordinados a través de sincronización explícita. Esta forma de paralelismo es la más común en programas escritos usando hilos.

### ¿QUÉ ES LA COMPUTACION PARALELA?

El procesamiento paralelo es un tipo de procesamiento de la información, que permite que se ejecuten varios procesos concurrentemente [5, 10, 17, 30, 35]. El procesamiento paralelo puede ser de diferentes tipos: i) Un tipo es ejecutar procesos independientes simultáneamente, los cuales son controlados por el sistema operativo (usando tiempo compartido, multiprogramación y multiprocesamiento). ii) Otro tipo es descomponer los programas en tareas (controladas por el sistema operativo, los compiladores, los lenguajes de programación, etc.), algunas de las cuales pueden ser ejecutadas en paralelo. iii) Finalmente, el último tipo se basa en usar técnicas de encauzamiento para introducir paralelismo a nivel de instrucciones, lo que implica dividirlas en pasos sucesivos que pueden ser ejecutados en paralelo, cada uno procesando datos diferentes. Es difícil determinar qué resulta más natural, sí el procesamiento paralelo o el secuencial. Los enfoques paralelos resultan una necesidad, dada la constante afirmación de que la velocidad máxima en procesamiento secuencial se alcanzará prontamente, excepto que aparezcan avances en otras áreas que definan nuevos mecanismos de procesamiento, los cuales pueden venir de nuevos descubrimientos en áreas tales como computación cuántica

Gráfico, Gráfico de líneas, Gráfico de cajas y bigotes

Descripción generada automáticamente

## ¿COMO SE REALIZA LA MULTIPLICACION DE DOS MATRICES?

Dos matrices A y B son multiplicables si el número de columnas de A coincide con el número de filas de B.



El elemento C***ij*** de la matriz producto se obtiene multiplicando cada elemento de la fila i de la matriz A por cada elemento de la columna j de la matriz B y sumándolos.

Visualmente se puede tener que:

Imagen de la pantalla de un celular con texto e imágenes

Descripción generada automáticamente con confianza baja

## ¿QUÉ ES LA LEY DE AMDAHL?

La Ley de Amdahl es llamada así por el arquitecto de ordenadores Gene Amdahl, se usa para averiguar la mejora máxima de un sistema cuando solo una parte de éste es mejorada. Establece que: La mejora obtenida en el rendimiento de un sistema debido a la alteración de uno de sus componentes está limitada por la fracción de tiempo que se utiliza dicho componente. El incremento de velocidad de un programa utilizando múltiples procesadores en computación distribuida está limitada por la fracción secuencial del programa

Gráfico, Gráfico de líneas

Descripción generada automáticamente

La Ley de Amdahl nos da una visión pesimista de las ventajas de la paralelización ya que nos dice que tenemos limitada la escalabilidad, y que este límite depende de la fracción de código no paralelizable.

La fórmula original de la ley de Amdahl es la siguiente:



siendo:

Fm el tiempo de ejecución mejorado • Fa el tiempo de ejecución antiguo. • Am el factor de mejora que se ha introducido en el subsistema mejorado.

Con la formulación de Amdahl el porcentaje de mejora en el rendimiento de un procesador viene limitado por el porcentaje del impacto del elemento que se quiere modificar. La ley de amdahl se generalizando es aplicable a cualquier problema que tenga una parte mejorable y otra que no se pueda mejorar.

Para nuestra investigación aplicaremos los principios de la ley de Amdahl de manera empírica y por medio de observación siguiendo la siguiente forma:

Texto

Descripción generada automáticamente con confianza baja

Donde nuestro speed up rate será igual a el tiempo de ejecución secuencial divido el tiempo de ejecución paralela

SPR = serial execution / parallel execution

## ¿QUÉ ES HIGH PERFORMANCE COMPUTING?

La computación de alto rendimiento (HPC) representa la capacidad de procesar datos y realizar cálculos complejos a velocidades muy altas. Para ponerlo en perspectiva, un equipo portátil o de sobremesa con un procesador de 3 GHz puede realizar unos 3.000 millones de cálculos por segundo. Aunque esto es mucho más rápido de lo que puede lograr cualquier humano, palidece en comparación con las soluciones HPC que pueden realizar cuadrillones de cálculos por segundo.

Uno de los tipos de soluciones HPC más conocidos es el superordenador. Un superordenador contiene miles de nodos de computación que trabajan juntos para completar una o varias tareas. Esto se denomina procesamiento paralelo. Es similar a tener miles de equipos conectados en red, combinando la potencia informática para completar tareas más rápidamente.

## ¿QUÉ SON LOS HILOS Y LA CONCURRENCIA DE MULTIPLES HILOS?

De la misma manera que un Sistema Operativo puede ejecutar varios procesos al mismo tiempo bien sea por concurrencia o paralelismo, dentro de un proceso puede haber varios hilos de ejecución. Por tanto, un hilo puede definirse como cada secuencia de control dentro de un proceso que ejecuta sus instrucciones de forma independiente. En la Figura puede verse cómo sobre el hardware subyacente (una o varias CPU’s) se sitúa el Sistema Operativo. Sobre éste se sitúan los procesos (Pi) que pueden ejecutarse concurrentemente y dentro de estos se ejecutan los hilos (hj) que también se pueden ejecutar de forma concurrente dentro del proceso. Es decir, tenemos concurrencia a dos niveles, una entre procesos y otra entre hilos de un mismo proceso. Si por ejemplo tenemos dos procesadores, se podrían estar ejecutando al mismo tiempo el hilo 1 del proceso 1 y el hilo 2 del proceso 3. Otra posibilidad podría ser el hilo 1 y el hilo 2 del proceso 1.

Los procesos son entidades pesadas. La estructura del proceso está en la parte del núcleo y, cada vez que el proceso quiere acceder a ella, tiene que hacer algún tipo de llamada al sistema, consumiendo tiempo extra de procesador. Por otra parte, los cambios de contexto entre procesos son costosos en cuanto a tiempo de computación se refiere. Por el contrario, la estructura de los hilos reside en el espacio de usuario, con lo que un hilo es una entidad ligera. Los hilos comparten la información del proceso (código, datos, etc). Si un hilo modifica una variable del proceso, el resto de los hilos verán esa modificación cuando accedan a esa variable. Los cambios de contexto entre hilos consumen poco tiempo de procesador, de ahí su éxito.

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

# MARCO CONTEXTUAL

Para la realización del presente caso de estudio se utilizaron las instalaciones personales para este caso, en el cual se implementó y ejecuto las pruebas necesarias para la recolección de datos, con un tiempo estimado de 4 horas, que suman el proceso total, desde la creación de los programas necesarios para multiplicar las matrices de forma secuencial y paralela, la ejecución y recolección de datos. Este tiempo parcial de 4 horas fue utilizado para la parte final de la fase investigativa, pues en total se consideran unas 4 horas adicionales de aprendizaje, descritas a continuación.

En un principio se optó por realizar el desarrollo del programa para la multiplicación de matrices en el lenguaje de programación Python, por comodidad de conocimiento en el desarrollo de dicho lenguaje del integrante del grupo Oscar Giraldo, así se llevó a cabo la implementación de la multiplicación de matrices de forma secuencial la cual arrojaba tiempos largos de ejecución de CPU que se consideraban normales en su momento, aquí un ejemplo:

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Posteriormente se realizó el desarrollo del programa de multiplicación de matrices de forma paralela utilizando procesos, el cual arrojaba tiempos de respuesta más cortos, lo esperado para este proyecto, pero erróneo, pues se debía implementar una solución que utilizara hilos y no procesos, Rápidamente se buscó la solución para aplicar hilos a la multiplicación de matrices de forma paralela en Python, y se encontró una solución, Dicha solución no presentaba mejoras en los tiempos de ejecución respecto a la implementación de forma secuencial lo cual genero duda y controversia sobre si se encontraba bien implementada, o si algo externo estaba creando estos tiempos largos en tiempo de ejecución, por lo que investigando se encontró que: en Python cuando de concurrencia hablamos, se trata del candado global del intérprete de Python o “Global Interpreter Lock (GIL)” es un método para el manejo de memoria que usa Python, lo cual durante el desarrollo que se planteó en el modelo paralelo, no se notaron mejoras en los tiempos de eje durante cada prueba que se hacía para la recolección de datos, esto se debe a que Python es un lenguaje interpretado; por lo que ocasiona este “GIL” es que solo permite que un hilo corra al tiempo en el intérprete de Python, lo que significa que solo un hilo se puede estar ejecutando en un momento dado, lo que ocasiona que Python no fue una muy buena elección para los casos de estudio de paralelismo.

Sumado a lo anterior, uno de los compañeros de clase de un equipo de trabajo diferente, presento quejas en las cuales mencionaba a Python y la dificultad que tuvo su equipo al desarrollar en este lenguaje pues tampoco encontraban mejoras de desempeño en sus pruebas. Por lo que optamos inmediatamente de buscar una solución en el lenguaje de programación C, logrando implementar tanto la multiplicación de matrices secuencial como la paralela y encontrando de forma muy directa una gran diferencia en los tiempos de ejecución secuenciales entre ambos lenguajes de programación Python vs C, y notando mejoras al momento de implementar la multiplicación de matrices de forma paralela utilizando el lenguaje de programación C con la librería PThread.

Tiempos de ejecución secuenciales Python vs C

Tiempos de ejecución secuenciales Python

Texto

Descripción generada automáticamente

Tiempos de ejecución secuenciales C

Texto

Descripción generada automáticamente

# DESARROLLO

# PRUEBAS