

Comparativo de rendimiento entre 3 ordenadores con python y cython sobre la ecuación de calor

Nicolás Cifuentes Barriga

Escuela de Ciencias Exactas e Ingeniería

Universidad Sergio Arboleda

Bogotá, Colombia

nicolas.cifuentes01@correo.usa.edu.co

Santiago Gutiérrez Orjuela

Escuela de Ciencias Exactas e Ingeniería

Universidad Sergio Arboleda

Bogotá, Colombia

santiago.gutierrez02@correo.usa.edu.co

Daniel Julian Siachoque Peralta

Escuela de Ciencias Exactas e Ingeniería

Universidad Sergio Arboleda

Bogotá, Colombia

daniel.siachoque01@correo.usa.edu.co

Juan Guillermo Torres Delgado

Escuela de Ciencias Exactas e Ingeniería

Universidad Sergio Arboleda

Bogotá, Colombia

juan.torres01@correo.usa.edu.co

Abstract— In the following laboratory report, the elaboration of a program based on the equation of heat, execution and respective performance time based on cython and python will be presented, in order to compare the execution speed between both and to study the per what of the same. On the other hand, it should be noted that the execution test of 3 different computers will be taken, evaluating in a more thorough way the performance of the program. Finally, the results given with the executions would be analyzed and represented in order to get a better understanding of them.

Keywords—Cython, Heat equation, Performance, Python, Graphic, dimension, compiling, translation.

I. INTRODUCCIÓN

En el transcurso del curso de computación paralela y distribuida se han encontrado diferentes formas para evaluar el rendimiento de un programa, permitiendo esto realizar mejoras constantes a los mismos y adaptarlos de una forma más eficaz a nuestras necesidades como programadores.

En esta ocasión se realizará un programa basado en la ecuación de calor, la cual se entiende como una ecuación diferencial que es capaz de describir el cómo la temperatura es capaz de variar en un espacio con base al tiempo, la ecuación de calor se puede escribir como:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \quad (1)$$

Para entender la ecuación de una manera más simple es necesario saber que $u(x, y, t)$ comprende el campo de la temperatura que varía y α es la constante de difusividad térmica. Por otra parte, la forma de solución numérica abarca 2 dimensiones, discretizando el primer laplaciano como se observa a continuación [[1]].

$$\begin{aligned} \nabla^2 u = & \frac{u(i-1, j) - 2u(i, j) + u(i+1, j)}{(\Delta x^2)} \\ & + \frac{u(i, j-1) - 2u(i, j) + u(i, j+1)}{(\Delta y^2)} \end{aligned} \quad (2)$$

Recordemos que la condición inicial está dada por $u(t = 0) = u_0$ y que la única forma donde la ecuación es estable se encuentra en:

$$\Delta t < \frac{1}{2\alpha} \frac{(\Delta x \Delta y)^2}{(\Delta x)^2 + (\Delta y)^2} \quad (3)$$

Un término importante que es necesario recalcar es “Cython”, el cual es un compilador estático que optimiza tanto para el lenguaje de programación Python como para el lenguaje de programación Cython extendido, con esto se logra que el escribir las extensiones de C necesarias sean tan sencillas como las mismas del python [[2]].

Adicionalmente, Cython cuenta con la combinación de la potencia de Python y C, dichas funciones ayudaran a:

- Escribir código de python en C desde cualquier punto y que esté a su vez sea utilizado de forma nativa.
- Utilizar la depuración combinada a nivel de código fuente para encontrar errores en el código.
- La implementación de Numpy para interactuar eficazmente con grandes conjuntos de datos.
- Integrarse de forma nativa con datos existentes de bibliotecas y aplicaciones heredadas.

Al igual que todos estos términos otro que resalta es python pues es un lenguaje de programación multiparadigma, pues soporta parcialmente la orientación a objetos, al igual que la programación imperativa y, aunque en menor medida, también soporta la programación funcional. Algunas de las características de este es que es fácil de utilizar, cuenta con una gran cantidad de librerías y bibliotecas, además permite extenderse fácilmente y utilizarse desde un entorno interactivo.

Ahora bien, para poder llevar a cabo la conversión de python a cython es necesario comenzar colocando en la cabecera del programa:

```
#cython: language_level=3
cimport cython
```

Seguido de esto la creación del html con el cual se va a lograr observar en qué partes del código se requiere agregar cython, la manera en la que se puede determinar esto, es mediante las líneas de color amarillo oscuro las cuales muestran una menor optimización para la generación de dicho pdf se emplea el código (cython -a mycode.pyx), una vez se ha determinado esto se comienza a cambiar el código, ya sea agregando bibliotecas de c con `cdef extern` o definiendo variables y funciones con `cdef` donde se inicializan [3]].

Por otro lado, encontramos el término “.dat” que se refiere a los archivos de datos genéricos, dichos archivos pueden ser codificados en formato de texto plano, mientras que algunos archivos DAT se implementan con especificaciones de codificación binaria. Cabe resaltar que los archivos .dat sólo pueden ser puestos en marcha por ciertas aplicaciones [4]].

Otro aspecto importante del cual se hizo uso y referencia en el programa son el número de cores que se posee en cada máquina de los desarrolladores, siendo estos los que leen las instrucciones y ejecutan las acciones específicas, todo con el fin de analizar cómo es el rendimiento de ejecución variando la cantidad de núcleos utilizados para la misma.

Las memorias ram de cada máquina, las cual se encarga de almacenar las instrucciones para que sea posible acceder más rápido a las mismas, cabe recalcar que existen diferentes capacidades para la Ram como:

- 2 GB - Contenidos sencillos y trabajos básicos.
- 4 GB - Multi tareas moderadas y trabajos complejos.
- 8 GB - Multi tareas altas y trabajo de cualquier tipo.
- 16 GB - Multi Tareas exigentes y trabajos intensivos.

Cabe recalcar que actualmente hay muchas más capacidades disponibles, que siguen aumentando el rendimiento tanto en tareas cotidianas, juegos o contenidos multimediales de mucha exigencia.

II. MATERIALES Y EQUIPOS

Para el desarrollo de esta práctica se hace uso de los siguientes equipos:

- PC1: Nvidia Geforce Rtx 3070 8gb gddr6, Amd Ryzen 5600x / 12threads 4.6gz, chipset b450, 32gb ram 3600mhz, iceberg 700w 80+, 1tbs Sata, 512Gb Nvme con 1800 mbps de lectura(Solido).
- PC2: Nvidia Geforce GTX 1650 4gb GDDR6, Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, motherboard B360M DS3H, 16gb ram 3000mhz, corsair 175r rgb, 240Gb XPG Solido, 480Gb sata(Solido).
- PC3: intel(R) Core(TM) i5-6200U CPU @2.30GHz, 1 phisical processor, 2 cores, 4 threads.
- Software Visual studio code.
- Sistema operativo ubuntu 20.04
- Sistempa operativo windows 10 con terminal de ubuntu.

Los materiales necesarios para la simulación son:

- Fichero heat_main.py.
- Fichero heat.py.
- Fichero cy_heat.pyx.
- Fichero setup.py.

- Fichero bottle.dat.
- Fichero bottle_medium.dat.
- Fichero bottle_large.dat.
- Fichero makefile.

III. METODOLOGÍA

Con base a la información descrita en la sección I se procede a crear el desarrollo pertinente para una comparación de rendimiento entre los lenguajes de programación python contra cython sobre la ecuación de calor para botellas con tres dimensiones diferentes. Inicialmente se parte de los ficheros otorgados los cuales son: *heat_main.py*, *heat.py*, *bottle.dat*, *bottle_medium.dat* y *bottle_large.dat*. Los ficheros con formato ‘.dat’ contienen las dimensiones que presenta la botella tenidas en cuenta para analizar siendo la una con repeticiones mayores a la previa, pasando de 200 – 528 – 1057. Por su parte, el archivo main se encarga de presentar los cálculos de tiempo que toma la ejecución para la generación del mapa de calor. En base a estos se plantea la elaboración de los ficheros pertinentes para alterar el archivo heat, el cual contiene las funciones necesarias para el cálculo de la ecuación de calor, creando así un *cy_heat* con la traducción apropiada para el funcionamiento en cython.

Una vez desarrollado el archivo ‘.pyx’ con lenguaje cython es necesario crear un setup para una buena compilación del mismo, estos ficheros se pueden apreciar en el [repositorio](#), y un makefile con el cual se facilita la compilación y eliminación de los archivos producidos post la misma. Las alteraciones aplicadas sobre el main para hacer evaluación de ambos lenguajes se basa en iteraciones para cumplir con cada uno de los lenguajes y ficheros de dimensión. Del mismo indole se crean las imagenes con los respectivos mapas de calor y la duración en compilación para llegar a estos.

Una vez concluido el proceso de elaboración de ficheros se compila y ejecuta obteniendo los tiempos de ejecución para cada una de las compilaciones, estos son almacenados para preparar el análisis sobre los mismos. Así mismo, las imágenes resultantes se comparan entre el inicio y los resultados del mapa de calor para cada una respectivamente. Este proceso se repite con los 3 ordenadores presentados en la sección II Con los resultados de tiempo se procede finalmente a apreciar asertivamente la comparativa y hacer un análisis de la misma haciendo uso de ley de Amdahl apreciando las diferencias entre componentes y cómo estas se pueden relacionar con los resultados obtenidos.

Es importante tener en cuenta los procesos que se desarrollan en cada uno de los ordenadores a la hora de realizar la ejecución puesto que a pesar de que esta no tiene un nivel de complejidad como podría llegar a serlo una multiplicación de matrices cuadradas de dimensiones en miles. Sin embargo, esto no discrimina la alteración en resultados por parte de procesos en ejecución así que se recomienda un cuidado adecuado para no quitarle transparencia al proceso.

IV. RESULTADOS

En base al procedimiento planteado en la sección III se obtienen diversos resultados los cuales son respectivamente analizados en la siguiente sección. Partiendo desde los resultados obtenidos por parte de los mapas de calor se obtiene la figura 1 la cual presenta los mapas de calores tras ejecución de python. Asi mismo, se presenta esta misma por parte de la transformación a cython en la figura 2. En las dos previamente mencionada se presentan de derecha a izquierda el estado base y el resultado obtenido.

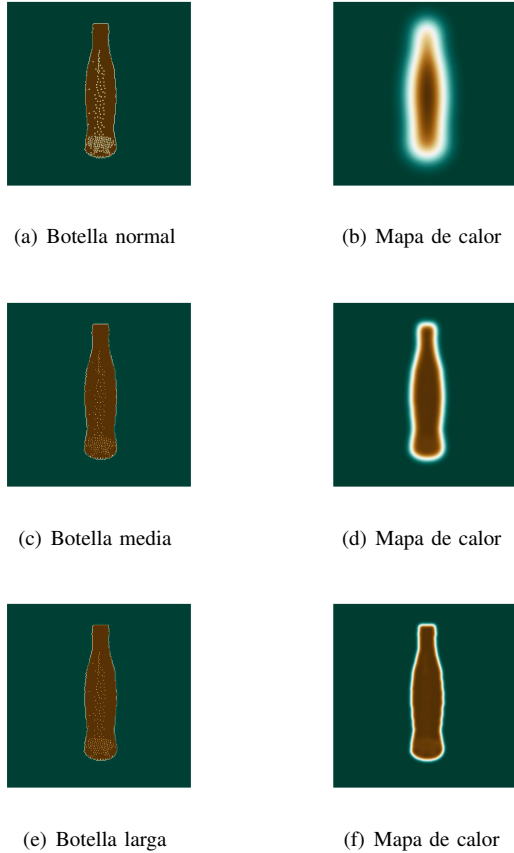


Figure 1. Resultados obtenidos por parte de la compilación con lenguaje python.

Una vez apreciados los resultados gráficos sobre la ecuación de calor se procede a almacenar la duración con cada una de las botellas y los respectivos lenguajes de programación. Lo previamente mencionado se presenta en la gráfica 3 para el PC1.

En base a los resultados de la imagen previamente presentada se procede a presentar una exclusivamente alusiva al resultado obtenido con el uso de cython la cual se aprecia en la figura 4 pudiendo asi verlo con mayor claridad.

Una vez obtenidos los mresultados por parte de cada uno de los ordenadores utilizados en la práctica se procede a apreciar como varia la duración entre cada ordenador dependiendo el lenguaje de programación y ordenador utilizado, obteniendo asi las figuras 5 y 6.

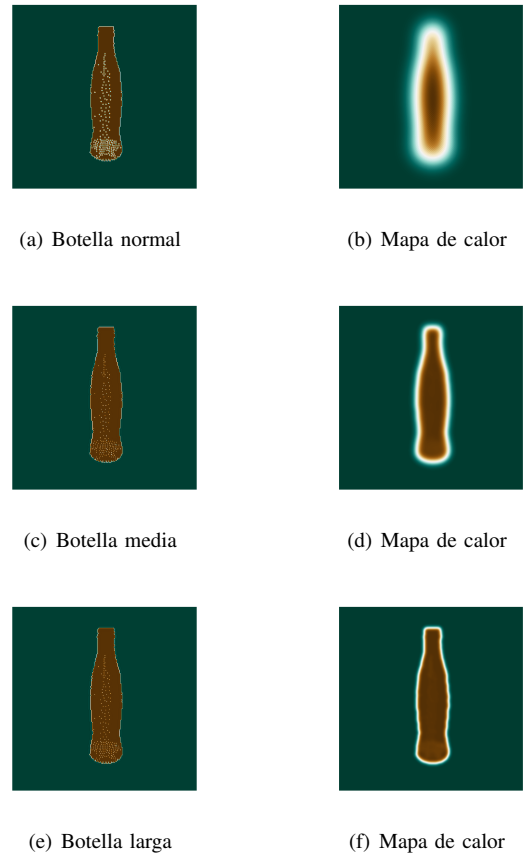


Figure 2. Resultados obtenidos por parte de la compilación con lenguaje cython.

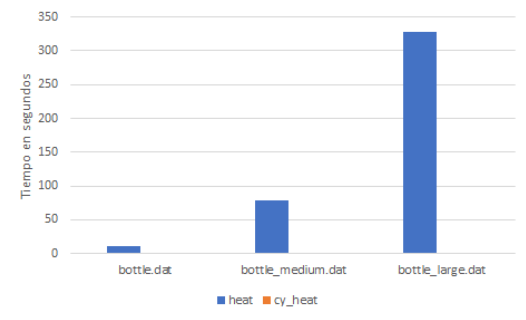


Figure 3. Gráfica resultante para la ejecución por parte del PC1 con python y cython.

Finalmente, con el fin de apreciar la diferencia entre cada ordenador y basandose en el lenguaje se aprecian las graficas 7 y 8 en las cuales se aprecia la duración para cda uno de los desarrollos establecidos.

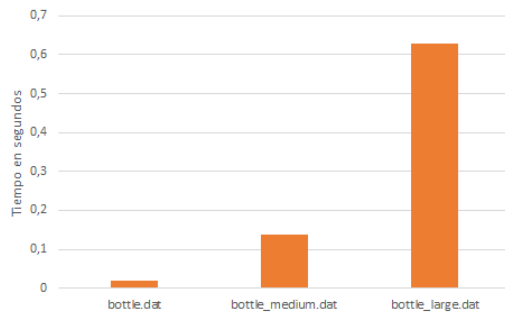


Figure 4. Gráfica resultante para la ejecución por parte del PC1 con cython.

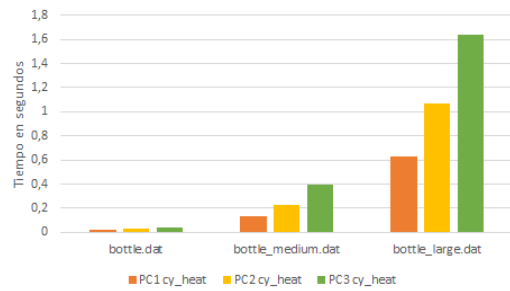


Figure 8. Gráfica resultante para la duración de ejecución entre los tres ordenadores de la práctica con el lenguaje cython.

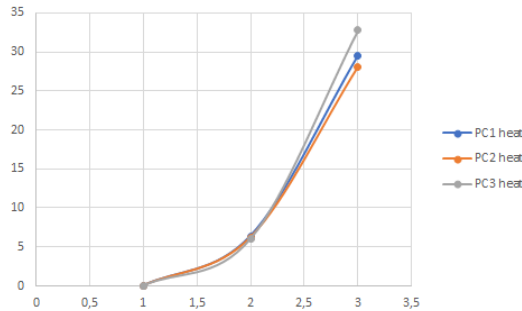


Figure 5. Gráfica resultante para el aumento en duración entre los tres ordenadores de la práctica con el lenguaje python.

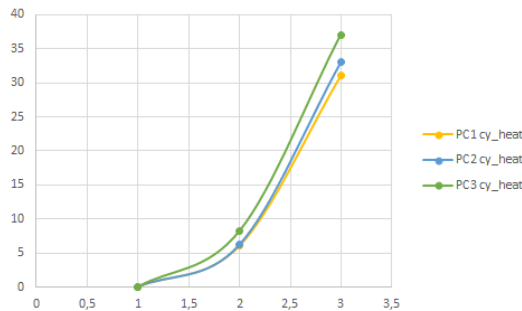


Figure 6. Gráfica resultante para el aumento en duración entre los tres ordenadores de la práctica con el lenguaje cython.

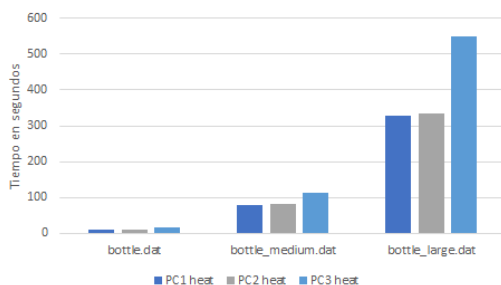


Figure 7. Gráfica resultante para la duración de ejecución entre los tres ordenadores de la práctica con el lenguaje python.

V. ANÁLISIS DE RESULTADOS

Podemos observar en la gráfica 1 y 2 que son iguales en cuanto al calor del objeto, a pesar de esto, en ambos se evidencia que a mayor tamaño del objeto, el calor en el estado final es menor.

Al observar la gráfica 3 podemos ver que la diferencia entre los tiempos de los programas de Python y Cython es abismal, tanto que no se observan los tiempos de ejecución de cy_heat, debido a esto se crean gráficas mostrando el tiempo solo de Cython, tal como en 4. Por siguiente, en las gráficas 5 y 6 se observa cuánto aumenta el tiempo de los programas con respecto al tamaño de las botellas, en ambos el aumento es similar a pesar de la diferencia de tiempo entre los programas, podemos ver que el tiempo de la botella media es apenas unas 6 veces mayor a la normal pero la larga tiene un tiempo de ejecución de al rededor de 30 veces más que el de la normal.

Las gráficas comparativas entre los tiempos de ejecución de los diferentes computadores se realizaron a parte las del programa en Python y las del Cython debido a la diferencia entre ellos antes mencionada, por parte de la gráfica 7 del archivo heat, observamos que la diferencia del tiempo entre PC1 y PC2 es casi nula en todos los tamaños de la botella mientras que el PC3 tiene un peor rendimiento, que se evidencia más en la botella larga, por otra parte, en la gráfica 8 el PC1 tiene el mejor rendimiento entre los 3 computadores, por siguiente el PC2 y por último el PC3, la diferencia entre el primero y el segundo es la similar a la que tiene el segundo y el tercero, además observando ambas, se puede ver la diferencia entre los programas que mencionamos, siendo el mayor tiempo en cy_heat del PC3 apenas un poco más de 1.6 segundos en la botella larga y el mejor tiempo en heat de más de 300 segundos del PC1, la mejora de rendimiento que tiene el programa al aplicarle Cython es demasiado notoria.

VI. CONCLUSIONES

- En base a las ejecuciones presentadas se puede apreciar como la biblioteca que consigue hacer unión entre los lenguajes de programación Python y C (Cython), consigue una optimización en la duración de desarrollo de los algoritmos en comparación al primero mencionado, siendo este una alternativa a tener en cuenta para optimización de programas en cuanto a rendimiento refiere.

- Se puede apreciar por parte de la ecuación de calor un comportamiento de carácter exponencial por parte de la duración de ejecución directamente proporcional a las repeticiones para las dimensiones de las botellas para el caso de este estudio.
- Los componentes del ordenador que realice la ejecución influye en los resultados obtenidos en duración a la ejecución, esta afirmación no sorprende puesto que se entiende que el nivel tecnológico naturalmente agiliza la duración de procesos.
- A pesar de ser un algoritmo relativamente ligero, se puede presentar afecciones en la pureza de los resultados en base a la cantidad de operaciones que este desarrollando el ordenador, por lo cual es importante mantener los recursos libres para la práctica y tener menor tasa de error en el producto final.

REFERENCES

- [1] J. Corredor, "Introducing heat equation," 2021.
- [2] P. Herron, "Learning cython programming," 2018.
- [3] S. Behnel, "Cython: C-extensionsfor python," 2010.
- [4] R. Soft, "Dat extension del archivo - ¿que es dat y como abrirlo?" 2017.

VII. AUTORES



Figure 9. Nicolás Cifuentes Barriga. Nació el 1 de noviembre del 2000. Actualmente cursa ingeniería en ciencias de la computación e inteligencia artificial junto con ingeniería electrónica. Tiene certificaciones y experiencia en campos de programación. Es cinturón negro de taekwondo y domina las lenguas español e inglés.



Figure 10. Santiago Gutiérrez Orjuela, oriundo de Villavicencio, Meta y nacido el 26 de agosto del 2000, actualmente estudiante de 7mo semestre de ingeniería de sistemas y telecomunicaciones con dominio de inglés.



Figure 11. Daniel Julián Siachoque Peralta, nació el 2 de octubre del 2000, en la ciudad de Bogotá. Actualmente cursa el 7mo semestre de ingeniería en ciencias de la computación e inteligencia artificial, curso diferentes programas de programación en python y bases de datos. Domina el español y el inglés.

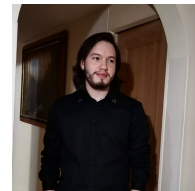


Figure 12. Juan Guillermo Torres Delgado, nacido el 10 de febrero del 2001 en Bucaramanga, Santander, estudiante de ingeniería en ciencias de la computación e inteligencia artificial, interesado en los temas seguridad de la información y las telecomunicaciones, cuenta con ciertos certificados en bases de datos y algunas en programación y cierta experiencia en el uso de programas de telecomunicaciones, ex jugador semi profesional de valorant y coach.