

## EA3: Evaluación de Aprendizaje Nro. 3.

### HPC (High Performance Computing o Computación de alto rendimiento)

Alumno: Jerónimo Fermín Deiros,

DNI: 28095186,

Comisión: 01-3900

Universidad Nacional de la Matanza,

Catedra Sistemas Operativos Avanzados (Sistemas Operativos – Plan '97),

[jdeiros@alumno.unlam.edu.ar](mailto:jdeiros@alumno.unlam.edu.ar)

**Resumen:** El objetivo de este documento es la realización de un informe para explicar de manera general el trabajo realizado para la Evaluación de Aprendizaje Nro. 3 de la materia Sistemas Operativos Avanzados, cuya documentación se encuentra detallada en cada uno de los 3 ejercicios que se van a entregar, realizados en la Plataforma Colab de Google.

#### **1 Resumen general de todo el trabajo**

Este trabajo tiene como objetivo principal aprender de manera practica sobre contenidos de la materia Sistemas Operativos Avanzados relacionados con la computación de alto rendimiento (HPC), pero también de manera practica obtener experiencia en el uso de la plataforma Colab de Google, que a su vez nos permite experimentar en la practica con computación de alto desempeño en GPU con CUDA (Arquitectura Unificada de Dispositivos de Cómputo) que es una plataforma que permite la programación GPGPU creada por NVIDIA que intenta aprovechar el paralelismo, y el alto ancho de banda de la memoria en las GPU en aplicaciones con un gran coste aritmético frente a realizar numerosos accesos a memoria principal. También se experimentó el uso de OpenMP que es una API (interfaz de programación de aplicaciones) para la programación multiproceso. Esto permitió la comparación de algoritmos realizados de manera secuencial y en paralelo, analizando las ventajas y dificultades de paralelismo. En el trabajo también se utilizaron lenguajes de programación como Python, C++ y C (adaptado para CUDA). Incursionando en el uso de cuadernos en Colab.

## 2 Explicación general del ejercicio 1

En este ejercicio se intenta replicar en el lenguaje Python, en la plataforma Colab, el algoritmo encontrado en el libro Como Programar en C C++ y Java de Deitel & Deitel, para la obtención de la moda, el valor mas frecuente entre un conjunto de valores. Para ello, se realizaron dos cuadernos Colab donde se procesarán en uno de manera secuencial y en otro en paralelo. Para la forma secuencial, utilizando el CPU, se carga un vector de N elementos, donde N se obtiene del usuario ingresando por parámetro su valor. La carga del vector se hace de manera aleatoria con valores que representan puntajes de 0 a 9, simulando la carga de datos de una encuesta. Luego, se utiliza un vector de frecuencias para contabilizar las ocurrencias, donde cada índice del vector representa los puntajes de 0 a 9 (10 valores posibles). Se procesan los resultados a través de un bucle for que recorre el vector resultados y contabiliza las apariciones de cada valor. Se capturan los tiempos de ejecución para poder comparar luego con la ejecución paralela utilizando GPGPU.

Para realizar el algoritmo en forma paralela se utilizo otro cuaderno de Colab, aquí se realizan la misma carga del vector de resultados de manera aleatoria, pero lo hace en forma paralela utilizando GPGPU. En este caso, el resultado no fue el esperado, se detalla en las conclusiones del ejercicio en el cuaderno de Colab los motivos, pero básicamente surgió un inconveniente para el algoritmo al contabilizar los resultados en el vector de frecuencia compartido por los hilos de procesamiento. Este ejercicio requirió de un análisis para evaluar el comportamiento del proceso y se tomo la decisión de dejar la solución propuesta, con el error de cálculos para evidenciar la situación. Mas allá del comportamiento en cuanto a los resultados, se realizó la comparación de los tiempos de ejecución con respecto a la versión secuencial.

### 2.1 URL a repositorio Git de Colab Ejercicio 1 - CPU

[https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros\\_Jeronimo\\_ejercicio\\_1\\_cpu.ipynb](https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros_Jeronimo_ejercicio_1_cpu.ipynb)

### 2.2 URL a repositorio Git de Colab Ejercicio 1 - GPU

[https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros\\_Jeronimo\\_ejercicio\\_1\\_gpu.ipynb](https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros_Jeronimo_ejercicio_1_gpu.ipynb)

## 3 Explicación general del ejercicio 2

El fin de los ejercicios 1 y 2 es comparar y analizar la ejecución en forma secuencial, solo usando CPU, y su versión paralela, optimizados con CUDA. En este ejercicio, como en el ejercicio 1, se crearon dos cuadernos Colab para realizar la comparación mencionada. En ambos cuadernos, se modifica el color de una imagen parametrizada, aplicando un filtro Sepia. En el caso secuencial se obtuvo una imagen a través de una URL, para esto se subió una imagen de prueba en el mismo repositorio Git utilizado para este trabajo y se convierte el formato comprimido JPG a un array para poder procesar píxel a píxel. Se procesa la imagen utilizando ciclos FOR para recorrer los pixeles y realizar los cálculos necesarios para aplicar el filtro sepia. En el algoritmo de procesamiento paralelo, a diferencia del ejercicio 1, en el ejercicio 2 el kernel ejecuta en hilos sobre dos dimensiones. Se definió una función kernel para ser ejecutada en GPU a través de la clase SourceModule() de CUDA. En ella se

realizan los cálculos para la transformación de la imagen aplicando el filtro Sepia en paralelo.

Comparando el procesamiento de la imagen con GPGPU contra la ejecución del ejercicio realizado en CPU con los ciclos for, se observa una disminución considerable de los tiempos de ejecución del algoritmo en GPU, siendo más óptimo el procesamiento con GPU, ya que, en este caso no tenemos la complejidad computacional ( $O$ ) que traen los for anidados ( $O^2$ ).

### 3.1 URL a repositorio Git de Colab Ejercicio 2 - CPU

[https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros\\_Jeronimo\\_ejercicio\\_2\\_cpu.ipynb](https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros_Jeronimo_ejercicio_2_cpu.ipynb)

### 3.2 URL a repositorio Git de Colab Ejercicio 2 - GPU

[https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros\\_Jeronimo\\_ejercicio\\_2\\_gpu.ipynb](https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros_Jeronimo_ejercicio_2_gpu.ipynb)

## 4 Explicación general del ejercicio 3

Para el ejercicio 3, se utilizó el mismo algoritmo del ejercicio 1, pero esta vez se hace en forma secuencial y en también en paralelo utilizando la biblioteca OMP en el lenguaje C, que corre en la plataforma Colab ejecutado desde Código Python. En las conclusiones del ejercicio se detalla los resultados con los tiempos obtenidos. Se llega a observar que dada la cantidad de núcleos reales que contamos para el procesamiento el nivel de paralelismo no se llega a apreciar tanto con los resultados óptimos.

### 4.1 URL a repositorio Git de Colab Ejercicio 3

[https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros\\_Jeronimo\\_ejercicio\\_3.ipynb](https://github.com/jdeiros/soa-2020/blob/master/HPC/Deiros_Jeronimo_ejercicio_3.ipynb)

## 5 Autoevaluación

Si bien creo que podría mejorar la originalidad de los ejercicios planteados, creo que esa es una parte complicada del trabajo ya que al momento de comenzar el estudio uno no está familiarizado con los conceptos y herramientas utilizados, con lo cual encontrar los algoritmos que mejor se adecuen y nos permitan comprender de la mejor manera los conceptos que estamos incorporando es una tarea que demanda tiempo de análisis, pero creo que es beneficioso para incorporar lo experimentado. Me resultó desafiante y me motiva a seguir investigando, pero por razones de tiempos decidí continuar con los ejercicios planteados, que a su vez creo que cumplen con el objetivo del trabajo en general.

Logré comprender los temas abordados, con respecto a la creación de programas utilizando GPGPU y openMP, analizando los beneficios y dificultades del paralelismo, también logré dominar las herramientas utilizadas.

Por otro lado, me esforcé en intentar cubrir no solo los puntos mínimos requeridos para la aprobación, sino que busqué apuntar a la promoción de la materia. Para ello, apliqué contenido teórico visto en la materia como OpenMP, organicé los cuadernos de Colab de manera de completar cada sección requerida: Introducción, armado de ambiente, desarrollo,

etc. También agregué excepciones en el Código para capturar los errores y, al menos informarlos, busque la forma para que todos los ejercicios fueran parametrizables y completé cada sección de Conclusiones en los ejercicios con lo solicitado. Creo que si tuviera que calificarme de acuerdo con los ítems propuestos para la calificación alcanzo la nota 9 (nueve).

De todas formas, lo mas importante, creo que es haber incorporado en un corto tiempo, la experiencia de un conjunto de herramientas y conceptos que seguramente estaré profundizando y que considero que son muy útiles para la profesión.