

# Descripción de la evaluación experimental de desempeño

## Objetivo

- Analizar el rendimiento de la implementación del algoritmo K-means en su versión serial y paralela. Además, buscamos determinar cómo varía el tiempo de ejecución promedio, dados los hilos utilizados y el número de puntos por clusterizar. Para comprobar si el algoritmo paralelo funciona, medimos el speed-up.

## Materiales y Métodos

- Datos de entrada: se utilizan conjuntos de datos en formato .csv de diferentes tamaños, representados por el número de puntos (100,000 - 1,000,000). Estos datos fueron generados con el código proporcionado en clase.
- Configuración de hilos: se hicieron las pruebas con 4 configuraciones de hilos (1,4,8,16).
- Medición de tiempo: se utilizó `opm_wget_time()` y `time.time()`.
- Análisis de datos: se automatizan las pruebas con ayuda de un script de python. Así mismo, se grafican y calculan los speedups para cada configuración.
- Para calcular los tiempos, se realizaron 10 iteraciones con cada configuración. Así buscamos obtener una medición de tiempo objetiva y atribuir los resultados al algoritmo en lugar de las condiciones del run. Al final, el tiempo usado en las gráficas y el análisis provienen del promedio de las 10 iteraciones.
- Descripción del equipo donde se ejecutaron los experimentos en términos de hardware y software.
  - Nombre del dispositivo: MarianaMoon
  - Procesador: 11th Gen Intel(R) Core(TM) i7-1195G7 @ 2.90GHz 2.92 GHz
  - RAM instalada : 16.0 GB (15.8 GB utilizable)
  - Tipo de sistema: Sistema operativo de 64 bits, procesador x64
  - Número de cores: 8, y 16 virtuales.
- Nombre del dispositivo: Mauricio-Vazquez-PC

Procesador: 10th Gen Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz

RAM instalada : 12.0 GB (11.8 utilizable)

Tipo de sistema: Sistema operativo de 64 bits, procesador x64

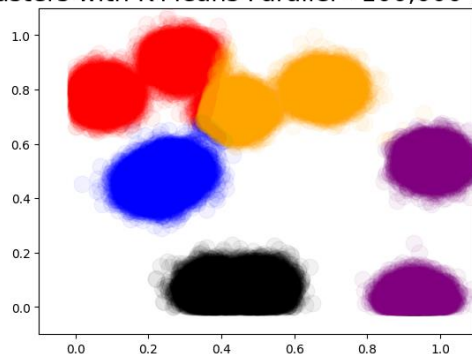
Número de cores: 4, y 8 virtuales.

- Software: utilizamos VScode.

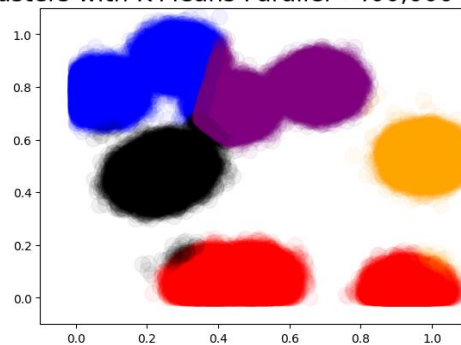
## Resultados y conclusiones

- En términos de funcionamiento algorítmico, nuestra versión de k-means logró clusterizar adecuadamente los puntos dados. En algunos casos observamos que la clusterización difiere un poco, se lo atribuimos al número de puntos en la prueba.
- Se observa que el tiempo de ejecución promedio disminuye significativamente al aumentar el número de hilos utilizados, especialmente para conjuntos de datos más grandes.
- Los resultados muestran que el speedup aumenta con el número de hilos, lo que indica una mejora en la eficiencia de la implementación paralela en comparación con la versión serial del algoritmo K-Means.
- Sin embargo, se observa una disminución en el speedup a medida que el número de puntos de datos aumenta, lo que sugiere posibles limitaciones en la escalabilidad de la implementación paralela.
- Además, observamos que hubo algunos picos en los cuales el speedup no supera el 1.5X.

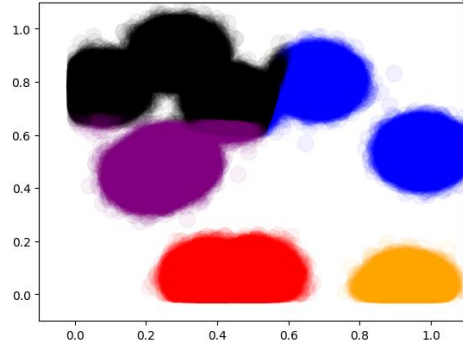
Clusters with K-Means Parallel - 100,000 points



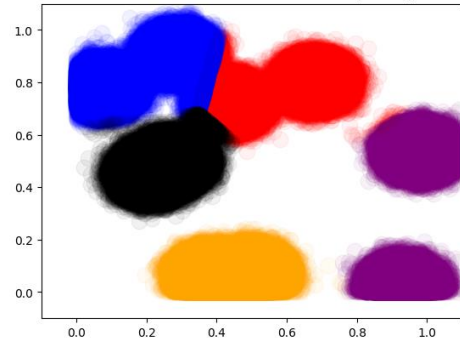
Clusters with K-Means Parallel - 400,000 points



Clusters with K-Means Parallel - 600,000 points



Clusters with K-Means Parallel - 1,000,000 points



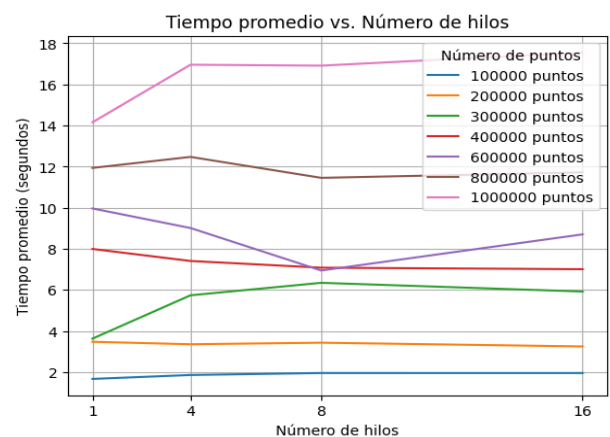
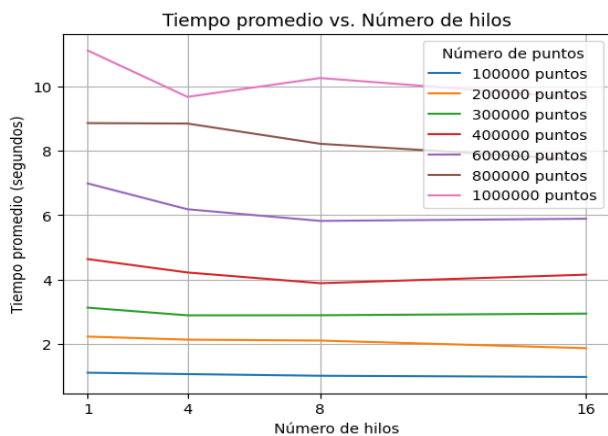
## Limitaciones y Áreas de Mejora

- Este experimento se realizó en un entorno específico (Windows) y con una implementación particular del algoritmo K-Means. Sería beneficioso realizar experimentos adicionales en diferentes entornos y con diferentes implementaciones para obtener una evaluación más completa del rendimiento.
- Se podría investigar el impacto de otros factores.

### Versión Paralela

vs.

### Versión Serial



### Gráfica de Speedups:

