Universidad tecnológica de Pereira
Informe sobre multiplicación de matrices

Presentado por:
Juan Pablo Peña M
Juan David Serna

Presentado a:
Gustavo Adolfo Gutierrez

Pereira, Julio de 2019
UTP

INFORME SOBRE EL ALGORITMO K-MEANS

Introducción

Se hace la implementación del algoritmo k-means que dentro de sus cualidades tiene reasignar clusters, calcular los centroides y la convergencia. Con distintos conjuntos de datos realizamos diferentes análisis con el fin de obtener grupos de resultados y a partir de estos verificar la calidad de la implementación y observar el desempeño del algoritmo, teniendo en cuenta la optimización del código y la implementación del mismo, esto nos dará una comparativa en cuanto a su compilación secuencial y paralela.

Análisis

Hacemos la implementación del código k-means y lo aplicamos a diferentes conjuntos de datos, a continuación se mostraran los resultados obtenidos a partir de su ejecución y además teniendo en cuenta el número de variables y la cantidad de instancias que posee cada DataSet.
Nota: Todos los tiempos se darán en milisegundos.

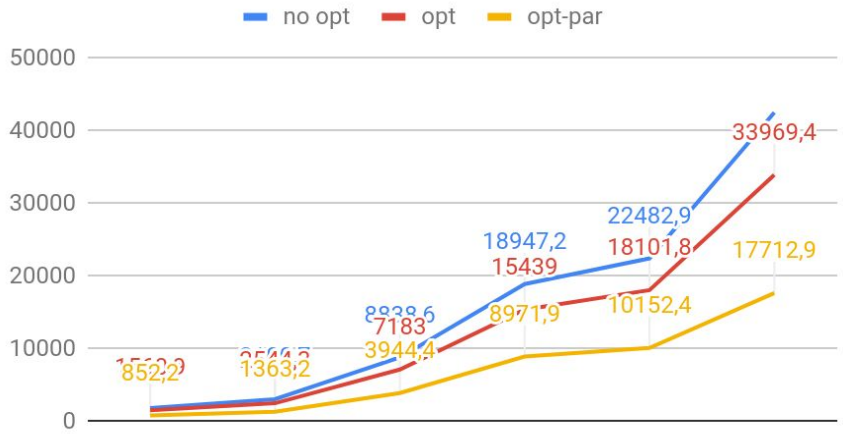
Dataset 1

- Características del dataset:
    - # de variables: 3.
    - # de instancias: 12969.
    - Descripción:

- Detalles de ejecución y pruebas:
    - # de ejecuciones para cada muestra: 10.
    - # de K utilizados {10,20,40,60,80,100} respectivamente.
    - Para todas las gráficas se utilizan los resultados promedio.
    - Se hace pruebas con 2 equipos distintos.

Para pc1: Intel® Celeron(R) CPU 1005M @ 1.90GHz × 2

caract\#k	10	20	40	60	80	100
no opt	1896,8	3100,7	8838,6	18947,2	22482,9	42552,1
opt	1560,9	2544,3	7183	15439	18101,8	33969,4
opt-par	852,2	1363,2	3944,4	8971,9	10152,4	17712,9

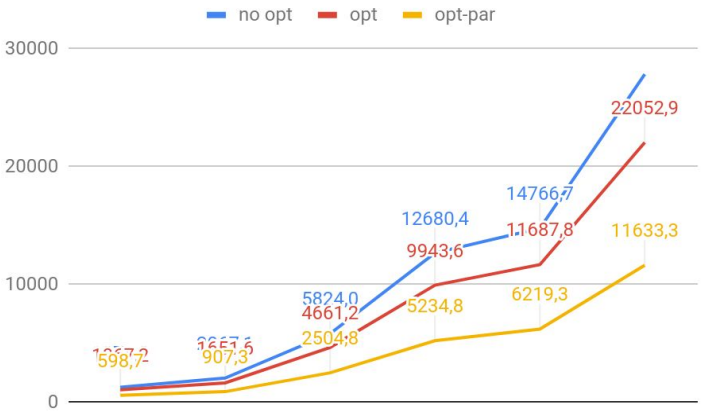


grafica 1

- Información:
  - % reducción “no opt” vs “opt”: 18,8%
  - % reducción “opt” vs “opt-par”: 45,1%
  - % reducción “opt-par” vs “no-opt”: 55,4%

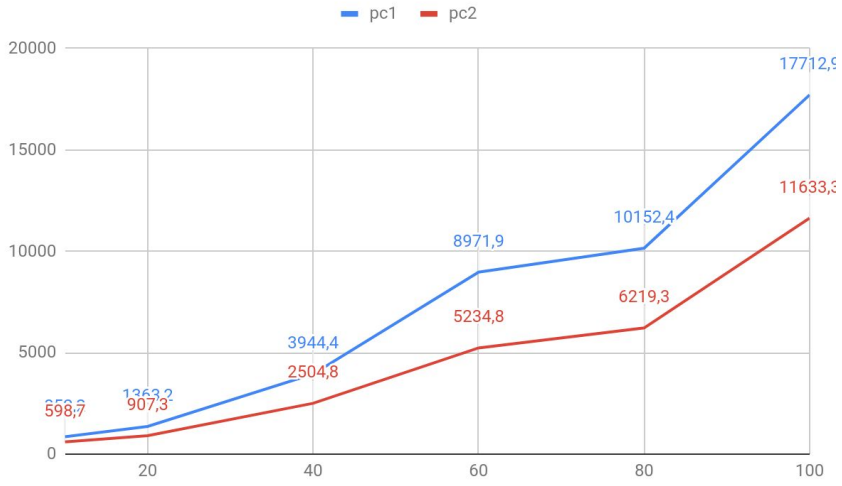
Para pc2: Intel® Core™ i5-4210U CPU @ 1.70GHz × 4

caract\#k	10	20	40	60	80	100
no opt	1274,0	2067,1	5824,0	12680,4	14766,7	27850,9
opt	1067,2	1651,6	4661,2	9943,6	11687,8	22052,9
opt-par	598,7	907,3	2504,8	5234,8	6219,3	11633,3



- Información:
    - % reducción “no-opt” vs “opt”: 19,9%
    - % reducción “opt” vs “opt-par”: 46,10%
    - % reducción “opt-par” vs “no-opt”: 56,82%

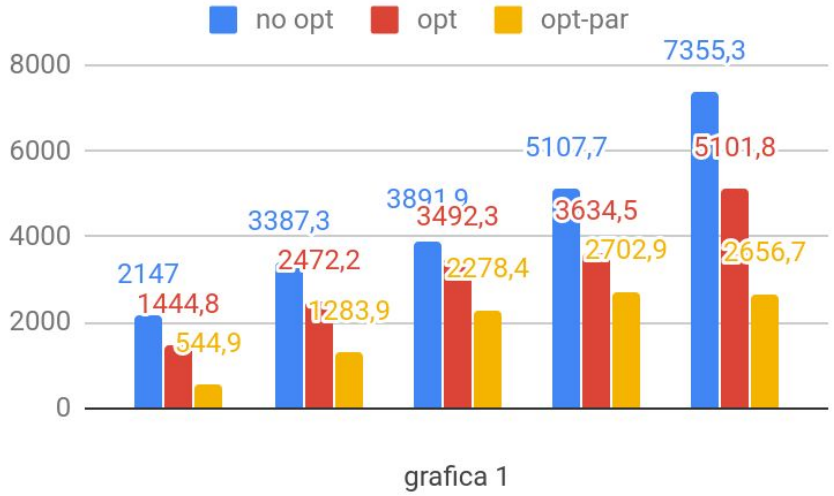
Comparativa entre 2 ambos equipos:



Se hace la comparativa utilizando el mismo dataset pero definiendo los valores iniciales de los k-means aleatorios.

Para pc1: Intel® Celeron(R) CPU 1005M @ 1.90GHz × 2

ntento\#k	20	40	60	80	100
no opt	2147	3387,3	3891,9	5107,7	7355,3
opt	1444,8	2472,2	3492,3	3634,5	5101,8
opt-par	544,9	1283,9	2278,4	2702,9	2656,7

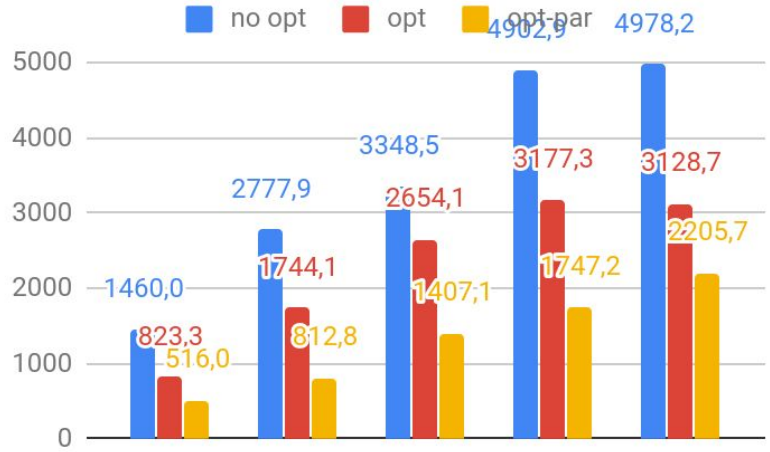


grafica 1

- Información:
    - % reducción “no-opt” vs “opt”: 25,9%
    - % reducción “opt” vs “opt-par”: 43,7%
    - % reducción “opt-par” vs “no-opt”: 57,8%

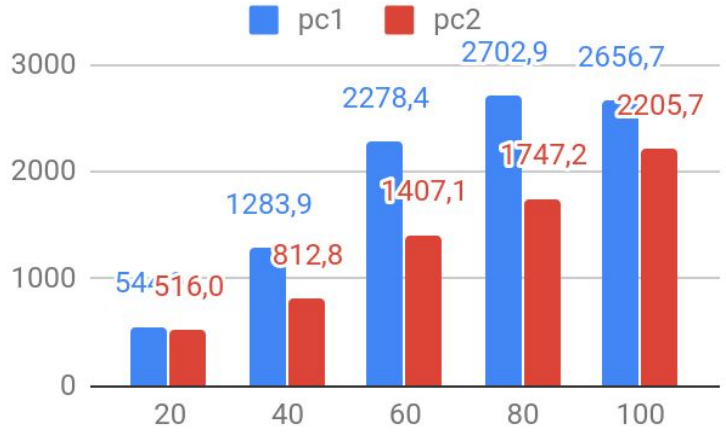
Para pc2: Intel® Core™ i5-4210U CPU @ 1.70GHz × 4

intento\#k	20	40	60	80	100
no opt	1460,0	2777,9	3348,5	4902,9	4978,2
opt	823,3	1744,1	2654,1	3177,3	3128,7
opt-par	516,0	812,8	1407,1	1747,2	2205,7



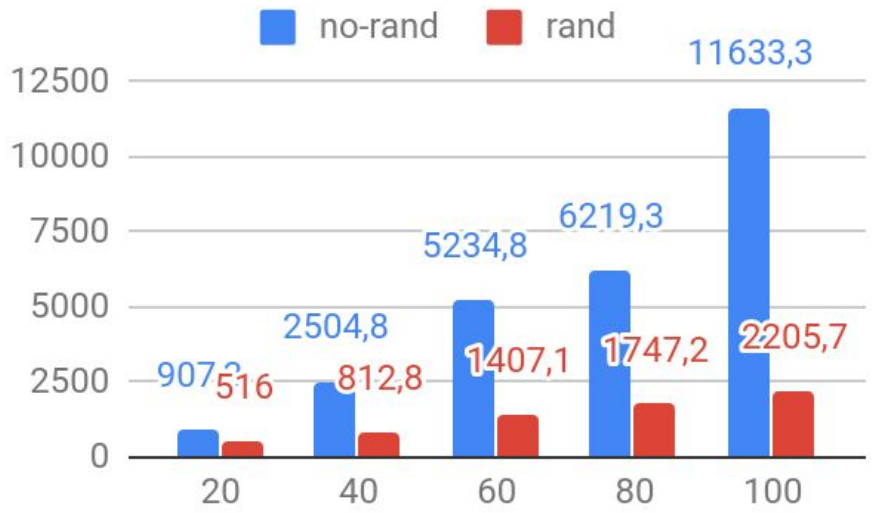
- Información:
    - % reducción “no-opt” vs “opt”: 34,8%
    - % reducción “opt” vs “opt-par”: 42,44%
    - % reducción “opt-par” vs “no-opt”: 62,69%

Comparativa entre 2 ambos equipos:



Comparativa de PC2 entre la ejecución optimizada paralela (random vs no-random):

pc\#k	20	40	60	80	100
no-rand	907,3	2504,8	5234,8	6219,3	11633,3
rand	516	812,8	1407,1	1747,2	2205,7
redu(%)	43,13	67,55	73,12	71,91	81,04



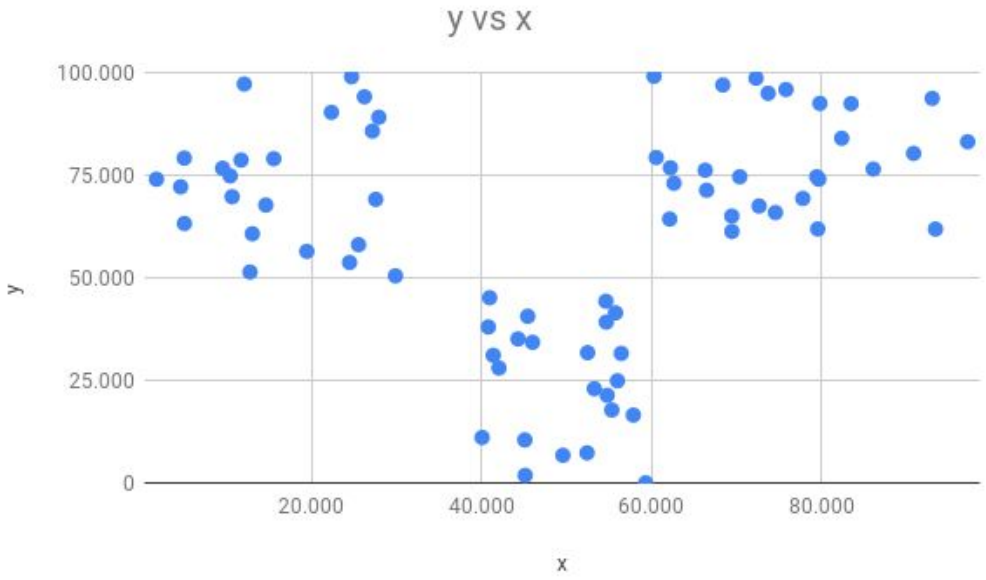
- Información:
    - Se hace la comparativa entre el comportamiento y disminución entre la asignacion aleatoria y no aleatoria

Pruebas y resultados

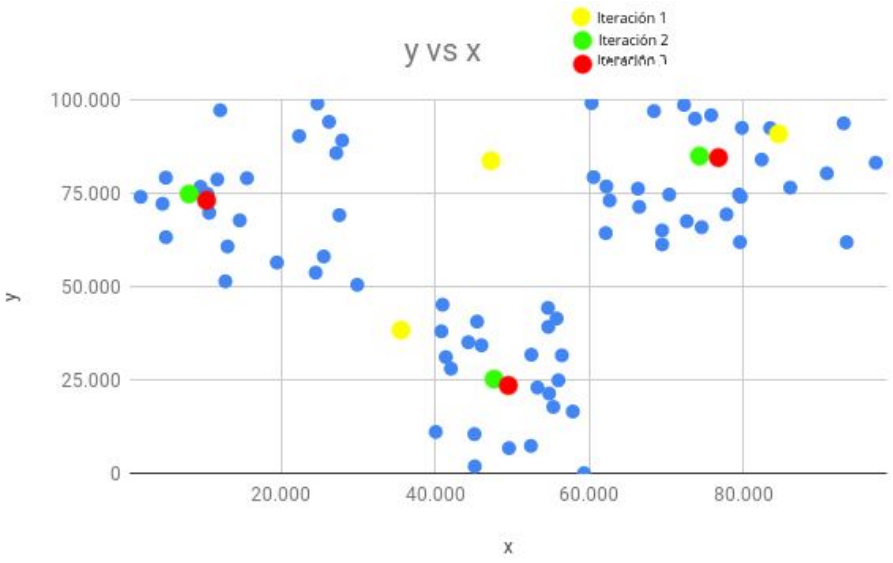
Dataset 2

- Análisis de prueba controlada
  Hacemos el análisis gráfico de un set de datos generado artificialmente con 3 tipos de datos genéricos, se ejecuta el algoritmo k-means para ese conjunto de datos.
    - Características del conjunto de datos:
        - Número de variables: 2
        - Número de instancias: 91
        - Tipo de datos: flotante con 3 decimales
        - Número de clusters(k): 3

- Número de iteraciones iniciales: 100
- Factor de convergencia: 0.000005
- Dominio de valores: x ∈ **R:** {0-100}
- Rango de valores: y ∈ **R:** {0-100}



- Resultados:
  Se ejecuta el algoritmo k-means, con valores iniciales de k-means aleatorios y ejecutado con el código de versión optimizada en el que el conjunto de datos se define en un solo vector y se obtienen e ingresan los resultados a la gráfica de valores, el tiempo de ejecución es muy bajo ya que el conjunto de valores es pequeño.



- Datos de ejecución:
    - Tiempo de ejecución: 0.0 ms.
    - Total iteraciones: 5.
    - Valores de k-means iniciales: Aleatorios.
    - La iteración 2 corresponde a la iteración número 3 en la ejecución real.
    - La iteración 3 corresponde a la iteración final en la ejecución real.

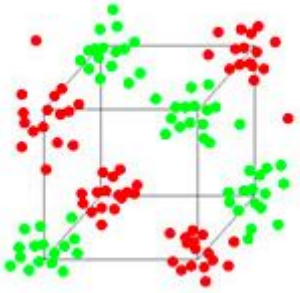
Dataset 3

- Información para dataset madelon_test.data :
    - Carácteristicas del dataset:
        - Número de variables: 500.
        - Número de instancias: 1800.
        - Tipo de datos: enteros positivos.
        - Número de clusters(k): {20,40,60,80,100}.
        - Número de iteraciones iniciales: 1000.
        - Factor de convergencia: 0.000005.

    - Descripción del dataset:

      MADELON is an artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1. The five dimensions

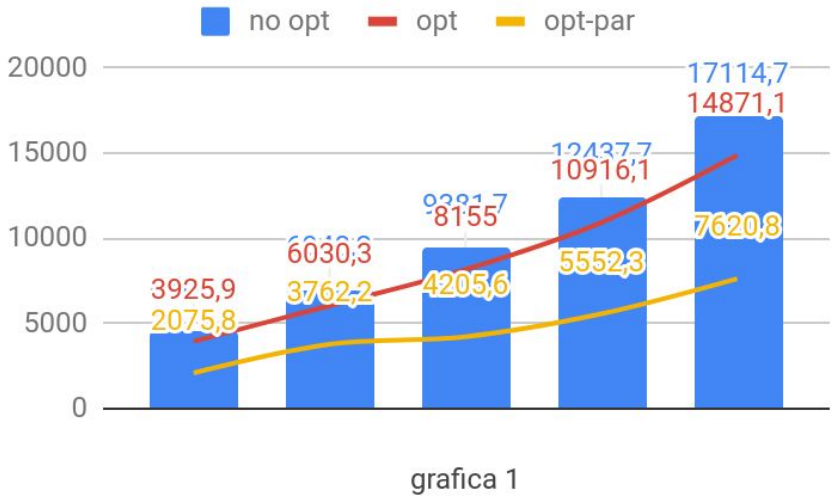
constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the 2 classes (corresponding to the +-1 labels). We added a number of distractor feature called 'probes' having no predictive power. The order of the features and patterns were randomized.



- Resultados:

Para pc1: Intel® Celeron(R) CPU 1005M @ 1.90GHz × 2

intento\#k	20	40	60	80	100
no opt	4543,8	6948,2	9381,7	12437,7	17114,7
opt	3925,9	6030,3	8155	10916,1	14871,1
opt-par	2075,8	3762,2	4205,6	5552,3	7620,8

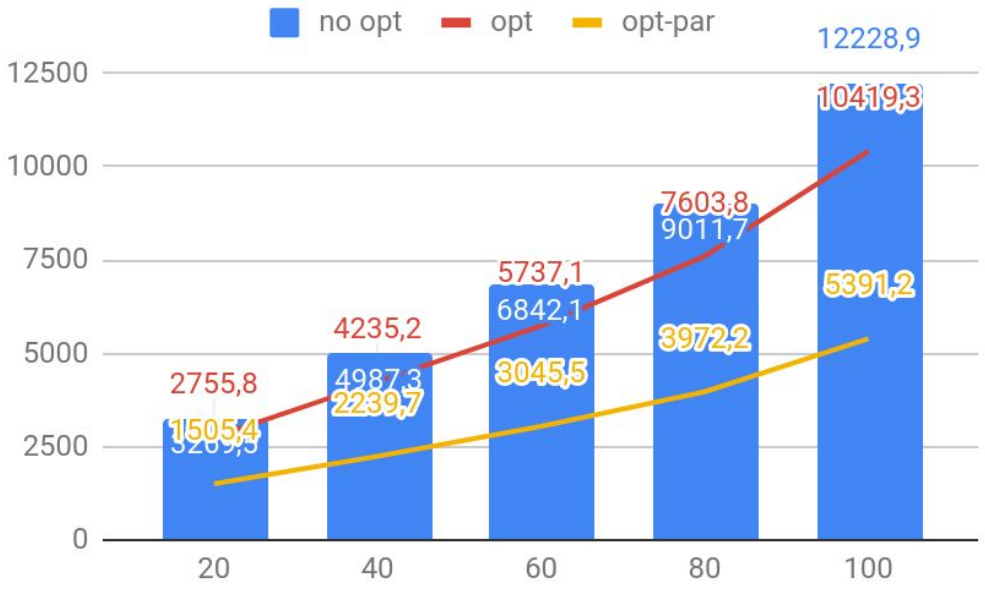


grafica 1

- Información:
    - % reducción “no-opt” vs “opt”: 13,0%
    - % reducción “opt” vs “opt-par”: 46,2%
    - % reducción “opt-par” vs “no-opt”: 53,2%

Para pc2: Intel® Core™ i5-4210U CPU @ 1.70GHz × 4

intento\#k	20	40	60	80	100
no opt	3269,5	4987,3	6842,1	9011,7	12228,9
opt	2755,8	4235,2	5737,1	7603,8	10419,3
opt-par	1505,4	2239,7	3045,5	3972,2	5391,2

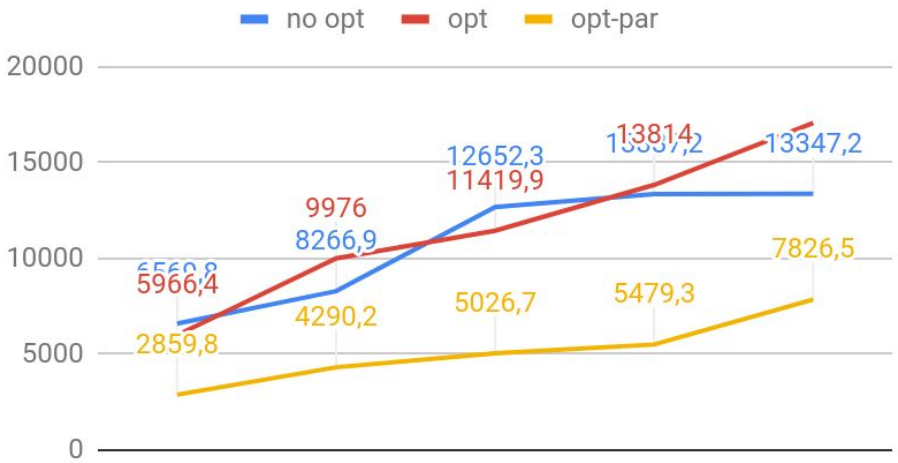


- Información:
    - % reducción “no-opt” vs “opt”: 15,5%
    - % reducción “opt” vs “opt-par”: 47,08%
    - % reducción “opt-par” vs “no-opt”: 55,27%

Se hacen las pruebas con el k-means inicial con valores aleatorios

Para pc1: Intel® Celeron(R) CPU 1005M @ 1.90GHz × 2

intento\#k	20	40	60	80	100
no opt	6569,8	8266,9	12652,3	13337,2	13347,2
opt	5966,4	9976	11419,9	13814	17045,5
opt-par	2859,8	4290,2	5026,7	5479,3	7826,5

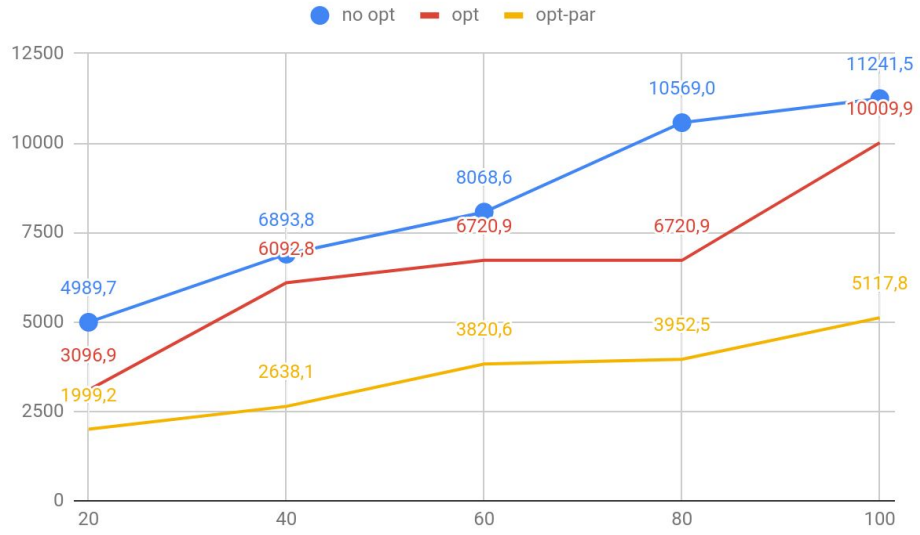


grafica 1

- Información:
  - % reducción “no-opt” vs “opt”: -6,6%
  - % reducción “opt” vs “opt-par”: 55,9%
  - % reducción “opt-par” vs “no-opt”: 53,0%

Para pc2: Intel® Core™ i5-4210U CPU @ 1.70GHz × 4

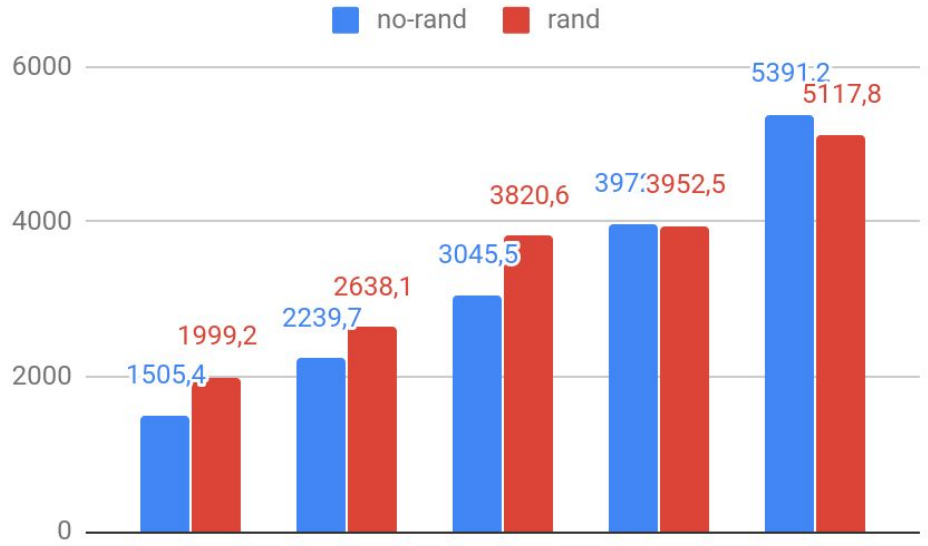
intento\#k	20	40	60	80	100
no opt	4989,7	6893,8	8068,6	10569,0	11241,5
opt	3096,9	6092,8	6720,9	6720,9	10009,9
opt-par	1999,2	2638,1	3820,6	3952,5	5117,8



- Información:
  - % reducción “no-opt” vs “opt”: 22,7%
  - % reducción “opt” vs “opt-par”: 45,07%
  - % reducción “opt-par” vs “no-opt”: 58,28%

Comparativa entre valores iniciales aleatorios y asignados:

pc\#k	20	40	60	80	100
no-rand	1505,4	2239,7	3045,5	3972,2	5391,2
desvesta1	25,0	12,1	95,1	26,9	40,8
rand	1999,2	2638,1	3820,6	3952,5	5117,8
desvesta2	342,5	350,9	855,9	516,4	867,7
redu(%)	-32,8	-17,8	-25,5	0,5	5,1



Comparación de porcentaje de reducción de tiempo entre los dataset 1 y 3

- Comparativa para los valores **no random**

pc\#k	20	40	60	80	100
dataset1	56,11	56,99	58,72	57,88	58,23
dataset3	53,96	55,08	55,49	55,92	55,91

- Comparativa para los valores **random**

pc\#k	20	40	60	80	100
dataset1	64,66	70,74	57,98	64,36	55,69
dataset3	59,93	61,73	52,65	62,60	54,47

Nota: todos los los valores en los campos corresponden a porcentaje (%) de disminución entre la ejecución secuencial no optimizada y la paralela

Conclusiones

Del algoritmo de K-means podemos evidenciar lo siguiente:

- Los resultados del análisis de la ***gráfica 1*** nos muestran la ejecución del algoritmo NO optimizado, el cual tiene como característica de diseño un conjunto de vectores los cuales están contenidos en otro vector.
  Con respecto a el algoritmo optimizado y el paralelo el NO optimizado presenta un rendimiento mucho menor al de los otros, realizamos unas serie de comparaciones para analizar el rendimiento de las tres versiones; evidenciamos que el algoritmo Optimizado tiene una reducción del 18,8% comparado al No optimizado y respecto a la versión en paralelo presenta una reducción del 55,4% frente a los tiempos que arrojó el algoritmo en su versión No optimizada .
- Se nota una reducción considerable en el tiempo debido a la ejecución en paralelo con respecto a la secuencial, además de notar una influencia del número de núcleos (cores) de la máquina en la que es ejecutado.
- Al obtener los resultados de la ejecución en la que los valores iniciales de los k-means están definidos aleatoriamente notamos que hay una desviacion estandar mayor de los datos y una reducción considerable en cuanto al desempeño, ya que el número de iteraciones para su convergencia es variable y puede disminuir con respecto a su ejecución con valores de clusters iniciales fijos, se nota una disminución hasta del 81,04%
- El algoritmo aumenta su complejidad directamente proporcional al número de k, también incrementa con respecto al número de variables

Webgrafía

- https://mockaroo.com
- http://archive.ics.uci.edu/ml/index.php
- https://arxiv.org/pdf/1312.4176.pdf
- http://archive.ics.uci.edu/ml/datasets/Madelon