Algoritmo de agrupamiento aplicado en la inteligencia de negocios usando GPGPU.

ISC. Misael Cosio Dominguez
Instituto Tecnológico de La Paz
ITLP
La Paz, Baja California Sur, México
M14310016@itlp.edu.mx

MSC. Iliana Castro Liera
Instituto Tecnológico de La Paz
ITLP
La Paz, Baja California Sur, México
icastro@itlp.edu.mx

MATI. Luis Armando Cardenas Florido
Instituto Tecnológico de La Paz
ITLP
La Paz, Baja California Sur, México
Armando.cardenas@itlp.edu.mx

MC. Jesús Antonio Castro
Instituto Tecnologico de La Paz
ITLP
La Paz, Baja California Sur, México
jcastro@itlp.edu.mx

DR. Marco Antonio Castro Liera
Instituto Tecnologico de La Paz
ITLP
La Paz, Baja California Sur, México
mcastro@itlp.edu.mx

Resumen – Se plantea una técnica de programación paralela (GPGPU) para la optimización del tiempo de ejecución de un algoritmo de agrupamiento en el procesamiento de información aplicado a la minería de datos, con el objetivo de encontrar un comportamiento de la información de una tienda de autoservicio.

Palabras Clave — Inteligencia de negocios, minería de datos, algoritmos de agrupamiento, K-means, GPU, programación paralela.

I. INTRODUCCIÓN

El procesamiento de grandes volúmenes de datos en un ambiente comercial y la interpretación de la información resultante requiere la utilización de herramientas computacionales de alto desempeño. La minería de datos aplicada a la inteligencia de los negocios[1] es una herramienta clave para encontrar factores importantes que permitan establecer el comportamiento de la información a través del tiempo y, a su vez, determinar un patrón entre los datos, con el fin de realizar una toma de decisiones de manera eficiente.

Los sistemas de información trabajan frecuentemente con grandes volúmenes de datos, y explotar ese volumen de información para llevar a cabo una toma de decisiones en un periodo de tiempo considerablemente corto tiene costos muy elevados para las organizaciones que utilizan hardware de capacidad limitada. El incremento del tiempo de procesamiento de la información se convierte en una gran limitante para obtener un mejor desempeño [2].

Una manera de mejorar el desempeño en el tiempo de procesamiento es la utilización de la técnica de procesamiento paralelo GPGPU (General Purpose Graphics Processing Unit).

Este proyecto se llevó a cabo en la División de Estudios de Posgrado e Investigación en el Instituto Tecnológico de La Paz. Los datos fueron proporcionados por la tienda de autoservicio Castores.

II. OBJETIVO

Paralelizar un algoritmo de agrupamiento utilizando técnicas de programación paralela con CUDA para comparar su eficiencia y eficacia contra resultados obtenidos tradicionalmente.

III. MATERIALES Y MÉTODOS

Para esta investigación se utilizó un algoritmo de minería de datos aplicado en la inteligencia de negocios, en combinación con las nuevas técnicas de programación paralela en una arquitectura CUDA.

El desarrollo de este proyecto se dividió en 3 secciones: la primera sección involucra los datos, en la segunda sección se trabajó el algoritmo de agrupamiento y, finalmente, el procesamiento paralelo.

A. Datos para la investigación

Para este proyecto se seleccionó una muestra experimental de 1, 000, 000 de registros de una base de datos de 19 millones de registros en SQL SERVER 2012. Cada registro posee 15 características del producto y a cada registro se le aplicó el proceso ETL (Extracción, Transformación y Carga) utilizado en la minería de datos como una herramienta que tiene como finalidad mantener una limpieza homogénea y coherencia entre los datos, para minimizar el margen error en su procesamiento.

Cabe mencionar que los registros seleccionados solamente comprenden las transacciones realizadas en un lapso de tiempo de 12 meses, en el Departamento de Licores del negocio.

ISBN: 978-607-97128-2-2

Cuando se trabaja con grandes cantidades de información, comprender los datos es un factor importante para el éxito en una toma de decisiones. Se creó un cubo multidimensional utilizando el complemento SQL DATA TOOL 2012 en VISUAL STUDIO 2012, para analizar de forma detallada los comportamientos que tienen los datos entre si y, además, visualizar las características que poseen cada uno de esos comportamientos. Los comportamientos se determinan a partir del funcionamiento del algoritmo de agrupamiento implementado en la minería de datos para la inteligencia de negocios.

B. Algoritmo de agrupamiento

Existe una gran variedad de algoritmos para la minería de datos o procesamiento de información para la inteligencia de negocios. En la tabla 1 se encuentran clasificados los algoritmos más utilizados en la minería de datos.

Nombre de algoritmos	PREDICTIVO		DESCRIPTIVO		
	Clasificación	Regresión	Agrupamiento	Reglas de asociación	Combinaciones / Factorizaciones
Redes neuronales	✓	✓	✓		
Arboles de decisión ID3, C4.5, C5.0	✓				
Arboles de decisión CART	✓	✓			
Otros árboles de decisión	✓	✓	✓	✓	
Redes de kohonen			✓		
Regresión lineal y logarítmica		✓			✓
Regresión logística	✓			✓	
K-means			✓		
A priori				✓	
Naive Bayes	✓				
Vecinos más próximos	✓	✓	✓		
Análisis factorial y de comp. ppales					✓
Twostep y Cobweb			✓		
Algoritmos genéticos y evolutivos	✓	✓	✓	✓	✓
Máquinas de vectores soporte	✓	✓	✓		
CN2 rules (cobertura)	✓			✓	
Análisis discriminante multivariante	✓				

Tabla 1.- Algoritmos implementados en minería de datos

Cada uno de estos algoritmos funciona de forma diferente, por lo que son tratados bajo condiciones controladas para problemas con un objetivo específico. Es difícil determinar o evaluar este tipo de algoritmos porque el desempeño que muestre un algoritmo depende de la problemática a resolver.

Para el desarrollo de esta investigación se trabajó con una técnica de agrupamiento (conocida en la literatura como clustering) cuyo objetivo es obtener grupos o conjuntos entre elementos de un universo de datos, de tal manera que los elementos asignados al mismo conjunto posean características similares.

En la tabla 1 se muestran los algoritmos descriptivos que trabajan con la técnica de agrupamiento. Dentro de esa clasificación se encuentra el algoritmo K-means. Este algoritmo es clasificado en la literatura como un algoritmo fácil de paralelizar [1].

K-means es un método de agrupamiento por vecindad en el que se parte de un número determinado de conjuntos y un determinado número de elementos por agrupar.

El objetivo de K-means es situar los K-centroides en el espacio, de forma que los datos más cercanos a determinado K-centroides formen una agrupación de datos a partir de una característica de similitud determinada por una función de distancia [1].

El desempeño del algoritmo K-means puede variar por la función distancia para un mismo problema. Una función distancia puede obtener mejores resultados que otra función distancia que se implemente al mismo algoritmo K-means, para el mismo problema a resolver.

Existen diferentes funciones de distancia que se pueden adaptar al algoritmo K-means; las más comúnmente usadas son:

- Distancia Euclídea

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(X_i - y_i)^2}$$

- Distancia de Manhattan

$$d(x,y) = \sum_{i=1}^{n}|X_i - y_i|$$

- Distancia de Chebychev

$$d(x,y) = max_{i=1\ldots n}|X_i - y_i|$$

- Distancia del Coseno

$$d(x,y) = arcos\left(\frac{X^T y}{(\|X\| \cdot \|y\|)}\right)$$

- Distancia de Mahalanobis

$$d(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$$

Estas funciones son las más conocidas, por ser de naturaleza sencilla y que trabajan con dos instancias X, Y. En un problema estas instancias representan dos características de un dato. Para la solución a nuestro problema se decidió implementar la función distancia euclídea, porque es la función más conocida y más sencilla de comprender. Además, la función distancia euclídea, por su definición, se basa en el concepto más común de distancia usado en el área de las matemáticas.

ISBN: 978-607-97128-2-2

C. *Procesamiento paralelo*

El procesamiento paralelo surgió de la necesidad de ejecutar múltiples procesos que permitan realizar una gran cantidad de operaciones en un periodo de tiempo muy pequeño.

CUDA es una plataforma de procesamiento paralelo y un modelo de programación creado por la compañía NVIDIA, que permite aprovechar al máximo el nivel de procesamiento que tienen las GPU NVIDIA [4].

Si una GPU tiene la capacidad de ejecutar un gran número de operaciones en un segundo, podemos realizar millones de operaciones simultáneamente en un grupo de GPGPU trabajando al mismo tiempo. Es por esto que al procesar grandes volúmenes de información utilizando las técnicas de procesamiento paralelo bajo la arquitectura CUDA se puede reducir de manera considerable el tiempo de procesamiento.

Para reducir el tiempo de procesamiento en esta investigación se desarrolló e implementó el algoritmo K-means en la arquitectura CUDA.

Se creó una estructura en la CPU que funciona como un arreglo para almacenar la información que se leyó del archivo fuente de texto plano. Una vez terminada la operación de lectura, se copió la información a una variable que se utilizada por la GPU. Posteriormente, se eligieron las características más relevantes y se aplicó una normalización a cada dato de la muestra para mantener los valores en rangos entre 0 y 1, para que los datos no presentaran valores muy altos y evitar un desplazamiento mayor de los K-centroides. Finalmente, se realizó el cálculo de la función euclídea con cada uno de los registros, utilizando cada uno de los K-centroides determinados inicialmente de forma secuencial.

IV. Resultados y conclusiones

Solo se ha considerado una prueba experimental para conocer el comportamiento de K-means en nuestro universo de datos en el Departamento de Licores.

Con el análisis detallado de SQL DATA TOOL, se obtuvieron los resultados mostrados.

Departamento Licores	
Total de registros	1,374,381
N° Clúster iniciales	10
% de similitud	100
Clúster con relación fuerte	clúster 2 y clúster 5

Tabla 2.- Información para el agrupamiento a 100% de similitud

En la tabla 2 se muestra la configuración del agrupamiento a 100 % de similitud, obteniendo una relación entre el clúster 2 y el clúster 5.

CLUSTER 2			CLUSTER 5		
Precio venta	0-45,7	99.781 %	Precio venta	0-45,7	100 %
Ventas	0-147,8	93.588 %	Ventas	0-147,8	100 %
Familia producto	JUGO/NECTAR	80.154 %	Familia producto	JUGO/NECTAR	72.297 %
Fecha compra	21-02-2013 al 27-09-2013	25.744 %	Fecha compra	21-02-2013 al 27-09-2013	25.778 %
	27-09-2013 al 03-05-2014	25.179 %		27-09-2013 al 03-05-2014	25.642 %
	03-05-2014 al 06-03-2015	18.955 %		03-05-2014 al 06-03-2015	19.429 %

Tabla 3.- Características de la relación en clúster 2 y clúster 5

La relación que tienen el clúster 2 y el clúster 5 está determinada por los valores que se presentan en las características de precio venta, ventas y familia del producto, como se muestra en la tabla 3.

Departamento Licores	
Total de registros	1,374,381
N° Clúster iniciales	10
% de similitud	80
Clúster con relación fuerte	clúster 2 y clúster 5 clúster 6 y clúster 7

Tabla 4.- Información para el agrupamiento a 80% de similitud

Aplicando una configuración a 80 % de similitud entre los agrupamientos, como se muestra en la tabla 4, se encontró otro agrupamiento de la relación, el cual se muestra la tabla 5.

CLUSTER 6			CLUSTER 7		
Precio venta	45,7 – 131,8	73.127 %	Precio venta	131,8- 428 ,5	59.856 %
Ventas	0-147,8	91.070 %	Ventas	147,8 – 608,8	75.164 %
Familia producto	VINO	33.983 %	Familia producto	VINO	28.272 %
Fecha compra	21-02-2013 al 27-09-2013	24.397 %	Fecha compra	21-02-2013 al 27-09-2013	22.534 %
	27-09-2013 al 03-05-2014	23.135 %		27-09-2013 al 03-05-2014	19.270 %
	03-05-2014 al 06-03-2015	18.196 %		03-05-2014 al 06-03-2015	14.838 %

Tabla 5.- Características de la relación en clúster 6 y clúster 7

Las métricas más importantes que se muestran en la tabla 3 y la tabla 5 son: precio de ventas, familia producto y fechas de compra.

La diferencia entre ambas tablas es que en la relación que muestra la tabla 3 se encuentra fuertemente relacionada la métrica familia-producto con el valor de JUGO/NECTAR, mientras que en la tabla 5 se encuentra relacionada la misma métrica, pero con el valor de VINO.

ISBN: 978-607-97128-2-2

Por lo anterior, podemos deducir que el grupo presentado en la tabla 3 realiza un mayor número de ventas con productos JUGO/NECTAR en ciertos intervalos de tiempo especificados, mientras que el agrupamiento de la tabla 5 muestra un mayor porcentaje de ventas con productos de VINO, en el mismo intervalo de tiempo.

El punto de partida para esta investigación es el análisis detallado que se obtuvo del SQL DATA TOOL con los datos procesados en cada una de las tablas de nuestra base de datos. Las agrupaciones obtenidas en este trabajo fueron comparadas con las agrupaciones obtenidas en la versión paralela, a fin de validar el funcionamiento del algoritmo K-means en la arquitectura CUDA.

Se programó una versión lineal, donde se adaptó la función distancia euclídea en el algoritmo K-means, para contrastar con la versión paralela.

En la tabla 6 se muestra la cantidad de datos procesados con una primera versión paralela del algoritmo K-means, así como los tiempos obtenidos en una versión lineal con la misma cantidad de registros procesados comparado con la versión paralela.

ALGORITMO K-MEANS		
CANTIDAD REGISTROS	VERSION LINEAL TIEMPO	VERSION PARALELA TIEMPO
100	0.010 seg	0.093 seg
1,000	0.253 seg	0.119 seg
10,000	5.790 seg	1.890 seg
1’000,000	17m39.646 seg	3m47.996 seg

Tabla 6.- Resultados obtenidos en tiempo de la versión lineal y versión paralela del algoritmo k-means

Los resultados de la tabla 6 muestran que paralelizar un algoritmo no siempre es lo más viable, ya que procesar pequeñas cantidades de información de forma paralela consume más tiempo en la parte de comunicación entre la CPU y la GPU que el tiempo que se tomaría la CPU en procesarlas.

Sin embargo, la versión paralela muestra una mejora del 80% que la versión lineal en el tiempo de ejecución al procesar 1, 000,000 de registros (como se describe en la tabla 6).

Es importante mencionar que una desventaja significativa al realizar operaciones con números decimales del lado de la GPU es la precisión, ya que una CPU tiene mayor precisión decimal que una GPU.

V. Trabajos Futuros

- Migrar la versión paralela del algoritmo K-means al clúster de GPU, utilizando MPI.
- Adaptar al algoritmo K-means otras funciones de distancias, para comparar el desempeño con la función euclídea aplicada al mismo problema.
- Probar con diferentes características, para determinar los agrupamientos en cada centroides determinado para el algoritmo K-means.

VI. Bibliografía

[1] Hernandez Orallo, J., Ramirez, M. J., & Ferri, C. (s.f.). *Introducción a la mineria de datos* . PEARSON.

[2] Pacheco, P. (s.f.). *An Introduction to Parallel Programming* . ELSEVIER -Morgan Kaufmann.

[3] Cook, S. (s.f.). *CUDA Programming - A developer´s guide to parallel computing with GPU's.* ELVESIER - Morgan Kaufmann.

[4] NVIDIA. (s.f.). *CUDA Parallel Computing Platform*. Obtenido de NVIDIA: http://www.nvidia.com/object/cuda_home_new.html

ISBN: 978-607-97128-2-2