Implementación de un Ambiente de Ciencia de Datos a través de la Librería Open Source Rapids

José Guarnizo1

1Departamento de Ciencias de la Computación y Electrónica  
Universidad Técnica Particular de Loja (Ecuador)  
[jaguarnizo4@utpl.edu.ec](mailto:jaguarnizo4@utpl.edu.ec)

Elizalde Rene2

1Departamento de Ciencias de la Computación y Electrónica  
Universidad Técnica Particular de Loja (Ecuador)  
rrelizalde@utpl.edu.ec

**Resumen.** La librería Open Source Rapids ejecuta canalizaciones de datos y análisis de extremo a extremo utilizando las principales características de las GPU, Rapids maneja grandes volúmenes de información en tiempo real y trabaja con computación acelerada, realiza una analítica de información en el proceso de extracción, transformación y carga e implementa conjuntos de algoritmos Machine Learning; se da realce a sus principales funcionalidades, características y su conjunto de bibliotecas de la librería, se utilizó la Plataforma Cloud BlazingSQL para la implementación, donde resalta la construcción de Ciencia de Datos para los procesos de ETL, Machine Learning y Visualización, se implementó una contraparte utilizando librerías que aún no tiene soporte para GPU como Pandas para ETL, para Machine Learning con Scikit-learn y para la visualización con Matplotlib, la comparación con las librerías tradicionales permite resaltar la aceleración que tiene Rapids cuando trabaja con grandes volúmenes de información.

**Palabras clave:** Ciencia de Datos. Rapids. GPU.

# Introducción

La cantidad de datos producidos en la actualidad son de gran volumen, es creciente en instituciones públicas y privadas, producen millones de datos al día y aportan considerables cantidades de información lo que causa que no tenga un adecuado tratamiento. Mediante un análisis completo de Rapids permitirá manejar grandes volúmenes de información en tiempo real y hacer uso de las características más importantes que permiten: ocupar el máximo de recursos disponibles, tiempos mínimos de cada proceso y capacidad para el análisis, procesamiento e implementación de algoritmos Machine Learning, con el interés de conocer, analizar, ampliar y utilizar la librería Rapids analizando un gran volumen de datos y que los tiempos de cada proceso sean rápidos.

# Librería Open Source Rapids

Rapids permite un procesamiento y capacitación enormemente acelerados en tamaños de conjuntos de datos mucho más grandes, brinda a los científicos de datos un enorme salto de rendimiento para resolver los desafíos comerciales más complejos como predecir fraudes con tarjetas de créditos, pronosticar inventarios de venta minorista y comprender el comportamiento de los clientes, etc.

El principal objetivo de la librería Rapids no es solo acelerar las partes individuales del flujo de trabajo típico de la ciencia de datos, sino acelerar el flujo de trabajo completo de extremo a extremo utilizando las características de GPU (Unidad de Procesamiento Gráfico), acelera su cadena de herramientas de ciencia de datos de Python con cambios mínimos de código, aumenta la precisión del modelo de Machine Learning iterando los modelos más rápido y es de código abierto. A continuación, se ira detallando algunas de las principales bibliotecas que tiene Rapids y que se empleó para la construcción en los procesos de ETL, Machine Learning y Visualización.

## Biblioteca Cudf

Cudf es la biblioteca de Rapids de manipulación de DataFrames basada en Apache Arrow, plataforma en varios idiomas, especifica un formato de memoria columnar estandarizado independiente del lenguaje para datos planos y jerárquicos, organizados para operaciones analíticas eficientes en hardware moderno (Arrow, 2016), enfocada en la preparación de datos, acelera la carga, el filtrado y la manipulación de datos (Aramburo, 2019), en otras palabras, realiza el proceso ETL (Extracción, Transformación y Carga).

## Biblioteca Cuml

Cuml es la biblioteca de Rapids para MachineLearning aceleradas por GPU, soporta los siguientes conjuntos de algoritmos: Clustering, Dimensionality Reduction, Linear Models for Regression or Classification, Nonlinear Models for Regresion or Classification, Time Series y permiten que el entrenamiento del modelo sea rápido.

## Biblioteca Cuxfilter

Cuxfilter es la biblioteca de Rapids para la visualización de los datos en forma gráfica, proporciona las conexiones entre diferentes bibliotecas de visualización y un marco de datos de GPU.

Cuxfilter resuelve los problemas al aprovechar el poder de la pila rapids.ai, principalmente cudf, los datos se mantienen en una GPU como un marco de datos y las operaciones como agregaciones grupales, clasificación y consulta se realizan en la propia GPU, solo devolviendo el resultado como salida a los gráficos (RAPIDS Development Team, 2018).

# Implementación de Rapids

La implementación de la librería se desarrolló en la plataforma BlazingSQL, acelera el procesamiento de datos y es construido en el ecosistema de Rapids. Aborda los gastos, la complejidad, el grande consumo de memoria en las CPUs y el ritmo lento (BlazingSQL, 2019). Los beneficios de utilizar BlazingSQL es que no necesita realizar una instalación de paquetes de Rapids, permite escribir, ejecutar código en Python en un navegador como Google Chrome (trabajo en Jupyter), la plataforma ya tiene preinstalada la librería, conjuntamente con otras librerías adicionales como pandas, dask, cupy, entre otras y tiene incorporado una GPU Tesla T4 esto permite poder optimizar el tiempo de trabajo y tomar decisiones para el proceso de construcción de ciencia de datos.

## Proceso de Extracción, Transformación y Carga (ETL)

La fase de extracción se trabajó con datos de OpenCampus (Comunidad global de aprendizaje, cursos abiertos), la data contiene 200.000 filas de información y con un tamaño de 252 MB. A continuación, el proceso de extracción y lectura de los datos.

Importar librerías

Importación de librerías para la implementación de Ciencia de Datos con Rapids.

Extracción y Lectura 

Extracción y lectura de los datos utilizando cudf.

**Fig. 1. Bibliotecas importadas de Rapids, extracción y lectura de la información con cudf.**

En la fase de transformación se procedió a realizar un análisis exploratorio de la información, seguidamente se eliminaron valores nulos, duplicados y vacíos, en la tabla 1 se detalla algunas características de cudf para la transformación de los datos.

**Tabla 1**. Características de cudf utilizadas para la transformación de los datos

|  |  |  |
| --- | --- | --- |
| Cudf | Descripción | Rapids |
| cudf.DataFrame.from\_pandas | Convertir de un dataframe de pandas a un dataframe cudf | Stable 0.14 |
| dropea | Eliminar filas con datos perdidos | Stable 0.14 |
| drop\_column | Eliminar columnas | Stable 0.14 |
| str.normalice\_spaces | Normalizar espacios vacíos | Stable 0.14 |
| str.split | Separar columnas | Stable 0.14 |
| columns | Incorporar nuevos nombres de columnas | Stable 0.14 |
| drop\_duplicates | Eliminar valores duplicados | Stable 0.14 |
| reset\_index | Resetear el index de forma ordenada | Stable 0.14 |
| join | Unir dataframes | Stable 0.14 |
| head | Información general de todo el dataframe | Stable 0.14 |
| shape | Dimensión del dataframe | Stable 0.14 |
| dtypes | Tipos de datos de cada columna del dataframe | Stable 0.14 |
| cudf.DataFrame | Crear un dataframe con cudf | Stable 0.14 |
| tolist | Convertir a una lista | Stable 0.14 |
| sort\_values | Ordenar por los valores en fila | Stable 0.14 |

Como los datos ya están transformados se procede a cargar a un nuevo archivo utilizando la característica de cudf de convertir todo el DataFrame (Marco de Datos) a archivo CSV, con esto finaliza el proceso de ETL.

Carga de datos

Carga de los datos a un nuevo archivo CSV utilizando cudf.

**Fig. 2. Carga de los datos.**

## Proceso Machine Learning (ML)

Dividir todo el Dataset (Archivo CSV) en Train y Test y convertir a datos categóricos escogiendo las columnas que se pretenden categorizar utilizando cuml.

Dividir Dataset

División del dataset en Train y Test utilizando cuml.

Preprocesamiento de los datos

Preprocesamiento de los datos utilizando cuml.


**Fig. 3. Dividir la data en Train, Test y convertir a datos categóricos.**

Para la aplicación de algoritmos Machine Learning se ha implementado dos tipos de algoritmos, Regresión Logística y Regresión Lineal, crea un modelo de entrenamiento de los datos trabajados (OpenCampus) para determinar la aceleración y rapidez de entrenamiento.

Regresión Lineal

Algoritmo regresión lineal de cuml.

**Fig. 4. Algoritmo Regresión Lineal de cuml.**

Regresión logística

Algoritmo regresión logística de cuml.

**Fig. 5. Algoritmo Regresión Logística de cuml.**

## Visualización de Gráficos

La visualización permite que Rapids presente los datos como un gráfico interactivo para facilitar la identificación y poder comprender conceptos difíciles. Para el proceso de visualización se construyó el gráfico estadístico de barras para representar los datos.

Construcción gráfico de barras

Construcción del gráfico de barras utilizando cuxfilter.


Visualización del gráfico

Representación gráfica de los datos en forma de barras.

**Fig. 6. Construcción y Gráfico de Barras.**

La visualización gráfica representa en el eje X la variable day y en el eje Y la variable username, por medio del gráfico representa el mayor día de conexión de los estudiantes a la plataforma OpenCampus para realizar la inscripción de los distintos cursos ofertados por la plataforma.

# Resultados de comparativa de aceleración de Rapids con las librerías tradicionales

En la tabla 2 se muestra la comparativa de aceleración (Tiempo en segundos) en el proceso de ETL utilizando Rapids y su contraparte Pandas.

**Tabla 2**. Comparativa de aceleración ETL con Rapids y Pandas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Librería | Fase | Tiempo Aceleración (segundos) | Número de Datos | Tamaño Archivo |
| Rapids (cudf) | Extracción | 0.70 s | 200.000 | 252 MB |
| Transformación | 7.24 s | 200.000 | 252 MB |
| Carga | 0.51 s | 200.000 | 252 MB |
| Pandas | Extracción | 3.17 s | 200.000 | 252 MB |
| Transformación | 8.14 s | 200.000 | 252 MB |
| Carga | 2.10 s | 200.000 | 252 MB |

En la tabla 3 se muestra la comparativa de aceleración (Tiempo en segundos) en el entrenamiento de los algoritmos ML utilizando Rapids y su contraparte Scikit-Learn.

**Tabla 3**. Comparativa de aceleración algoritmos ML con Rapids y Scikit-Learn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Librería | Algoritmo ML | Tiempo Aceleración (segundos) | Número de Datos | Tamaño Archivo |
| Rapids (cuml) | Regresión Lineal | 0.009 s | 157.493 | 41.6 MB |
| Regresión Logística | 16.42 s | 157.493 | 41.6 MB |
| Scikit-Learn | Regresión Lineal | 0.018 s | 157.493 | 41.6 MB |
| Regresión Logística | 484. 30 s | 157.493 | 41.6 MB |

En la tabla 4 se muestra la comparativa de aceleración (Tiempo en segundos) en el proceso de construcción y visualización de los datos de forma gráfica utilizando Rapids y su contraparte Matplotlib.

**Tabla 4**. Comparativa de aceleración para visualización de gráficos con Rapids y Matplotlib

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Librería | Tipo de Gráfico Estadístico | Tiempo Aceleración (segundos) | Número de Datos | Tamaño Archivo |
| Rapids (cuxfilter) | Barras | 4.25 s | 157.493 | 41.6 MB |
| Matplotlib | Barras | 191.22 s | 157.493 | 41.6 MB |

En base a los escenarios propuestos comparando los tiempos de aceleración de Rapids juntamente con las librerías tradicionales para Ciencia de Datos en los procesos ETL, Machine Learning y Visualización destaca Rapids, acelera el flujo de trabajo de extremo a extremo utilizando las características de las GPUs de NVIDIA, esto da un realce significativo para utilizar la librería Rapids cuando se trabaja con grandes volúmenes de información.

La aceleración y rapidez es parte fundamental de la librería Rapids, las pruebas y los resultados conformaron un análisis importante de aceleración en los procesos de ambientes de Ciencia de Datos, donde Rapids permite mejorar los tiempos de esfuerzo y trabajo de cada proceso como se detallaron en las tablas de resultados de aceleración, esto permitirá que los científicos de datos tomen decisiones importantes en cuanto a los datos analizados, entrenados y presentados para dar criterios de mejoría para las empresas, organizaciones, instituciones que necesiten dar un valor significativo de los datos y que mejor para acelerar el flujo de trabajo es utilizar Rapids.

# Conclusiones

Actualmente una parte clave de la Ciencia de Datos es la exploración de los datos para después preparar un conjunto de datos para entrenar con algoritmos de Machine Learning, a medida que crecen el conjunto de datos, la interactividad de este proceso se ve afectada cuando se ejecuta en CPU, concluyo que la Librería Open Source Rapids revela detalles de trabajo que adopta en ambientes de proceso exploratorio de datos, apoya en mejorar el futuro proceso de Ciencias de Datos para cosechar toda su potencia de trabajo y aceleración de las GPUs.

Rapids perfecciona el flujo de analizar, procesar, entrenar y solucionar problemas, brinda a los desarrolladores, ingenieros, científicos de datos trabajar de manera acelerada y que los tiempos de cada proceso ETL, Machine Learning y Visualización de los datos son rápidos cuando trabaja con grandes volúmenes de información, ayuda a que la interacción con los datos sea más sencilla, rápida, beneficioso y que permita sacar el mejor provecho de los datos, creo que con toda la Ciencia de Datos se acelerará por GPU en el futuro utilizando la librería Rapids.

# Referencias

Aramburo, R. (2019). BlazingSQL Parte 1: El GPU DataFrame (GDF) y cuDF en RAPIDS AI. *CuDF — GPU Data Processing for GDFs*, *0*(0). https://blog.blazingdb.com/blazingsql-part-1-the-gpu-dataframe-gdf-and-cudf-in-rapids-ai-96ec15102240

Arrow, A. (2016). *The Apache Software Foundation*. https://arrow.apache.org/

BlazingSQL. (2019). *blazingSQL*. https://www.blazingsql.com/

RAPIDS Development Team. (2018). *RAPIDS: Collection of Libraries for End to End GPU Data Science*. https://rapids.ai

m/rapids-ai/gpu-accelerated-cyber-log-parsing-with-rapids-10896f57eee9

Rees, B. (2019). *RAPIDS cuGraph*. https://medium.com/rapids-ai/rapids-cugraph-1ab2d9a39ec6

Rhodes, B. (2019). Análisis acelerado de GPU de Cyber ​​Log con RAPIDS. *Las Operaciones de Seguridad (SecOps) y Los Departamentos de TI Están Recopilando, Gestionando e Intentando Analizar Más Datos Que Nunca. Es Probable Que Los Empleados Conecten Sus Propios Dispositivos a Las Redes Corporativas, Ampliando Aún Más Una Super*. https://medium.com/rapids-ai/gpu-accelerated-cyber-log-parsing-with-rapids-10896f57eee9

Keller, C. A., Clune, T. L., Thompson, M. A., Stroud, M. A., Evans, M. J., & Ronaghi, Z. (2019). *Accelerated simulation of air pollution using NVIDIA RAPIDS*. *November*, 4–6.