International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 16 No. 2 Jun. 2016, pp. 378-389

http://www.ijias.issr-journals.org/

Evaluación del marco de trabajo Hadoop y Power View en la Visualización de Trayectorias GPS Vehicular

[Evaluation framework Hadoop and Power View display in GPS Vehicle Trajectories]

Gary Reyes Zambrano, José Alvarado Santos, Katia Villafuerte Ponce, Oscar Leon de La Torre, Fernando Coral Moran, and Vicente Arreaga Figueroa

Facultad de Ciencias Matemáticas y Físicas,
Universidad de Guayaquil,
Guayaquil, Ecuador

ABSTRACT: This article describes the evaluation of work Hadoop framework and complement Excel Power View through an experiment analyzing large volumes of information from GPS vehicle trajectories. In order to do a study to use Hadoop's own tools, USA dataset with information used trucks and their respective routes. This research was conducted in the following stages: 1) selection work environment where we see what are the best features and the need to work with Hadoop, 2) hardware to setup the environment and features for the analysis of GPS, 3) paths loading, analysis and visualization of results. Using Hive it is studied as a data store and the transformation of the tables to a format that facilitates ORC information processing. At the stage of data analysis it was used to perform MapReduce algorithms and PIG to make a risk assessment using SQL code conversions. Lastly displays and interprets the results with Power View a feature of Microsoft Excel 2013, which shows a map with GPS coordinates for all vehicles, where analysis techniques could conclude that 40% of accidents on the roads of EE California USA is caused by driver fatigue. For future work will proceed to generate GPS paths of the city of Guayaquil to determine patterns in their behavior.

KEYWORDS: Business, Intelligent, Hive, MapReduce, SQL.

RESUMEN: El presente artículo describe la evaluación del marco de trabajo Hadoop y del complemento Power View de Excel a través de un experimento de análisis de gran volumen de información de trayectorias GPS vehiculares. Con la finalidad de hacer un estudio que permita utilizar las herramientas propias de Hadoop, se utiliza un Dataset de EEUU con información de camiones y sus rutas respectivas. Esta investigación se desarrolló siguiendo las siguientes fases: 1) selección del ambiente de trabajo donde vemos cuales son las características óptimas y el hardware necesario para trabajar con Hadoop, 2) realizar la configuración del ambiente y características para el análisis de trayectorias GPS, 3) la carga, análisis y visualización de resultados. Se estudia el uso de Hive como almacén de datos y para la transformación de las tablas a un formato ORC que facilita el procesamiento de la información. En la etapa de análisis de Datos se usó MapReduce para realizar algoritmos y PIG para hacer un estudio de riesgos mediante conversiones de código SQL. Por último se visualiza e interpreta los resultados con Power View una característica de Microsoft Excel 2013, que muestra un mapa con todas las coordenadas GPS de los vehículos, donde mediante técnicas de análisis pudimos concluir que el 40% de los accidentes en las carreteras de California EE UU se ocasiona por la fatiga de los conductores. Para futuros trabajos se procederá a generar trayectorias GPS de la ciudad de Guayaquil para determinar patrones en su comportamiento.

KEYWORDS: Business, Intelligent, Hive, MapReduce, SQL.

Corresponding Author: Gary Reyes Zambrano

1 INTRODUCCIÓN

Winshuttle [1] asegura que el término «Big Data» se empleó por primera vez en un artículo de los investigadores de la NASA Michael Cox y David Ellsworth. Ambos afirmaron que el ritmo de crecimiento de los datos empezaba a ser un problema para los sistemas informáticos actuales. Esto se denominó "El Problema del Big Data". Desde entonces la capacidad para producir información ha aumentado vertiginosamente con respecto a algunos años atrás. El profesor de bioinformática de la Escuela de Salud Pública de Harvard, Winston Hide asegura que en los últimos cinco años se ha generado más información científica que en toda la historia de la humanidad [2]. El volumen de los datos existentes es de tal magnitud que, si ocupara un espacio físico, superaría el tamaño de una galaxia.

Actualmente las empresas generan grandes volúmenes de información, siendo este su activo más importante e indispensable para la toma de decisiones. Esto ha hecho necesario el desarrollo de herramientas que analicen y procesen los datos para identificar la información más relevante. Para el almacenamiento de la información se utilizan las bases de datos relacionales y Datawarehouse por sus características de capacidad de soporte, estrategia que ha dado muy buenos resultados en cuanto a la carga estructurada de datos, sin embargo el tiempo de respuesta en el procesamiento y análisis es solo aceptable.

En base a esta problemática es desarrollado Apache Hadoop, un Framework que ofrece almacenamiento fiable de datos mediante su sistema de archivos distribuidos DFS y el procesamiento de datos paralelo a través de algoritmos MapReduce. Hoy en día existen tecnologías y frameworks para Big data pero sus potencialidades apenas han comenzado a ser explotadas. Durante el desarrollo de este trabajo se tratará el uso Hadoop enfocado al procesamiento de datos, específicamente de un dataset con rutas vehiculares GPS. Es utilizado el dataset para realizar un análisis de los accidentes vehiculares y sus causas, que permite experimentar con la interfaz y diversos componentes del ecosistema Hadoop. El presente trabajo se estructura de la siguiente manera: en la sección 2 se analiza Bigdata con Hadoop en cuanto a herramientas y potencialidades asociadas a la investigación, en la sección 3 se analizan y discuten los resultados de la investigación y en la sección 4 se abordan las conclusiones y el trabajo futuro.

2 BIG DATA CON HADOOP

Hadoop es un framework que permite el procesamiento de grandes volúmenes de datos a través de clusters, usando un modelo simple de programación [3]. Este framework es reconocido además ç por su versatilidad en el manejo de gran cantidad de información, es muy utilizada por una gran variedad de desarrolladores que están en busca de herramientas que sean innovadoras y gratuitas. Manejar volúmenes de datos excesivamente grandes involucra tener un hardware con buenos recursos, por tal razón Hadoop está construido básicamente sobre dos módulos:

- Hadoop distributed file system (hdfs): el sistema de ficheros sobre el que se ejecutan la mayoría de las herramientas que conforman el ecosistema hadoop.
- Hadoop mapreduce: el principal framework de programación para el desarrollo de aplicaciones y algoritmos.

Aparte de estos bloques hay otras herramientas que completan el ecosistema Hadoop para desarrollar soluciones Big Data que pueden ser analizadas en [4].

Hive es una de las funcionalidades que ofrece Hadoop. En [5] se explica que posee una interfaz parecida a SQL la cual permite realizar toda clase de consultas en la base de datos de Hadoop. Es una herramienta para data Warehousing que facilita la creación, consulta y administración de grandes volúmenes de datos distribuidos en forma de tablas relacionales [6]. Esta funcionalidad cuenta con un lenguaje derivado de SQL, denominado Hive QL, que permite realizar las consultar sobre los datos. A su vez, Hive QL está construido sobre MapReduce, de manera que se sirve de las características de éste para trabajar con grandes cantidades de datos almacenados en Hadoop. Como principal limitación de esta funcionalidad se tiene que no ofrece respuestas en tiempo real.

La Figura1 muestra la arquitectura de Hive, que está compuesta de los siguientes componentes: [7]

- Interfaz de usuario: El método de entrada del usuario para realizar las consultas. Actualmente hay una interfaz de línea de comandos y una interfaz web.
- Driver: Recibe las consultas y se encarga de implementar las sesiones, además puede recibir consultas vía interfaces JDBC y ODBC.

- Compilador: Parsea la consulta y realiza análisis semánticos y otras comprobaciones de lenguaje para generar un plan de ejecución con la ayuda del metastore.
- Metastore: Almacena toda la información -metadatos- de la estructura que mantienen los datos dentro de Hive, tiene el esquema de las bases de datos, tablas, particiones, entre otros.
- Motores de ejecución: se encargan de llevar a cabo el plan de ejecución realizado por el compilador.

En resumen Hive es una herramienta familiar y altamente utilizada por los usuarios de base de datos, en gran medida por la gran similitud que posee con SQL. Hive realiza procesos importantes como son: resumen de los datos, consultas y análisis, ofreciendo herramientas que permiten una fácil extracción de datos, transformación y carga (ETL).

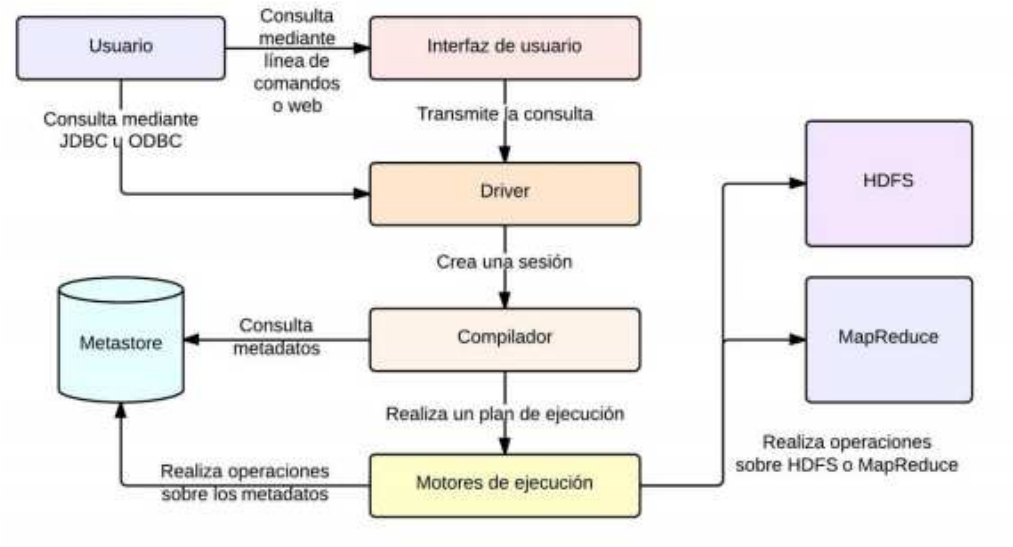


Fig 1. Arquitectura Hive. Tomado de [4]

ARQUITECTURA DEL SISTEMA DE DIRECTORIOS DISTRIBUIDO DE HADOOP: EL HDFS

El HDFS de Hadoop es un sistema de directorios que permite almacenar y gestionar cualquier tipo de datos, ya sean del tipo estructurados, semiestructurados o no estructurados, de manera distribuida en un clúster escalable de servidores [8]. La diferencia entre HDFS y otros sistemas de archivos consiste en que antes de comenzar el almacenamiento de la información cada archivo es fragmentado en bloques, cada fragmento es replicado el número de veces que el usuario especifique a través de nodos como denominados DataNodes como lo vemos en la figura 5. Esta arquitectura ha demostrado capacidad de ampliación de producción de hasta 200 MB de almacenamiento y un único conjunto de 4500 servidores, dando apoyo el apoyo a cerca de mil millones de archivos y bloques, además de proporcionar tolerancia a fallos distribuido en el almacenamiento [9]. A continuación se explica de manera detallada cada elemento de la arquitectura HDFS [10]:

Namenode

El namenode es el hardware básico que contiene el sistema operativo GNU/Linux y el software namenode. Es un software que puede ejecutarse en hardware. El sistema que tiene la namenode actúa como el servidor maestro y realiza las siguientes tareas:

- Administra el espacio de nombres del sistema de archivos.
- Del cliente regula el acceso a los ficheros.
- Ejecuta las operaciones del sistema de archivos como el cambio de nombre, cierre y apertura de archivos y directorios.

Datanode

La datanode es un hardware de productos básicos con el sistema operativo GNU/Linux y software datanode. Para cada nodo de un clúster, habrá un datanode. Estos nodos gestionan el almacenamiento de datos de su sistema. Los Datanodes realizar operaciones de lectura y escritura de los sistemas de archivos por petición del cliente. Permiten además realizar operaciones tales como creación, supresión, entre otros, con lo que la replicación de acuerdo con las instrucciones del namenode

Bloque

En general los datos de usuario se almacenan en los archivos de HDFS. El archivo en un sistema de archivos se divide en uno o más segmentos almacenados en los nodos de datos. Estos segmentos son denominados bloques y constituyen la estructura atómica para el almacenamiento de la información. La cantidad mínima de datos que HDFS puede leer o escribir se es un bloque. El tamaño de bloque por defecto es de 64 MB, pero puede ser aumentado por la necesidad de cambiar de configuración HDFS.

PIG

PIG es un lenguaje de programación de alto nivel que se utiliza con Hadoop. Este lenguaje permite a los trabajadores de datos escribir transformaciones de datos complejos sin conocer Java [5]. Fue diseñado para llevar a cabo una larga serie de operaciones de datos, por lo que es ideal para tres categorías de empleos Big Data:

- Extracción, transformación y carga (ETL) de tuberías de datos,
- Investigación en los datos en bruto, y
- Procesamiento de datos iterativo.

Además es una herramienta para analizar grandes volúmenes de datos mediante un lenguaje de alto nivel -PigLatin- que está diseñado para la paralelización del trabajo. Mediante el uso de un compilador se traducen las sentencias de PigLatin a trabajos MapReduce sin que el usuario tenga que pasar a programar ni tener conocimientos sobre las sentencias o el lenguaje de programación. De esta manera PigLatin es un lenguaje fácil para programar, pues se trata de un lenguaje textual y convierte las paralelizaciones en flujos de datos, conceptos más sencillos para los usuarios no expertos o sin conocimiento en el mundo de la paralelización. Además también se encarga de optimizar de manera automática los programas escritos por el usuario, respetando la coherencia de los datos, antes de traducirlos a trabajos MapReduce. Adicionalmente, permite la creación de funciones por parte del usuario para realizar procesamientos de propósito especial y que no se incluya en las operaciones básicas de PigLatin. [11]

Pig puede ejecutarse tanto en clústeres Hadoop -modo MapReduce- o en servidores sin una instalación Hadoop, sin ningún tipo de sistema distribuido -modo local-. Adicionalmente tiene dos modos para trabajar:

- Modo interactivo: se trabaja en un terminal grunt -una línea de comandos- y permite lanzar los comandos y sentencias una a una y de manera interactiva. Se lanza mediante el comando pig en una línea de comandos y con el argumento "-x" se le indica si se quiere lanzar en modo local o mapreduce.
- Modo batch: si se le pasa por argumento al comando "pig" un fichero de script, se ejecutará el dataflow que contenga el fichero. También acepta modo local y mapreduce.

MAPREDUCE

MapReduce es un modelo de programación introducido por Google y que en su evolución han participado decenas de colaboradores, apareciendo multitudes de implementaciones. De entre todas esas implementaciones destaca especialmente Hadoop, un proyecto de Apache para proporcionar una base sólida a las arquitecturas y herramientas Big Data. El objetivo de MapReduce es mejorar el procesamiento de grandes volúmenes de datos en sistemas distribuidos y está especialmente pensado para tratar ficheros del orden de gigabytes y terabytes. También mejora el tratamiento de los datos no estructurados, ya que trabaja a nivel de sistema de ficheros [4].

El nombre viene dado por la arquitectura del modelo, dividida principalmente en dos fases que se ejecutan en una infraestructura formada por varios nodos, formando un sistema distribuido, y que procede de la siguiente manera:

Map: uno de los nodos, con el rango de "master", se encarga de dividir los datos de entrada (uno o varios ficheros de gran tamaño) en varios bloques a ser tratados en paralelo por los nodos de tipo "worker map". Cada bloque es procesado independientemente del resto por un proceso que ejecuta una función map. Esta función tiene el objetivo de realizar el procesamiento de los datos y dejar los resultados en una lista de pares clave-valor.

$$(k1\,,\,v1\,) \rightarrow li(k2\,,\,v2)$$

Reduce: los nodos worker de tipo *reduce* ejecutan una función *reduce* que recibe como entrada una de las claves generadas en la etapa de *map* junto con una lista de los valores correspondientes a esa clave. Como salida genera una lista resultante de una función con los valores recibidos. La unión de los resultados puede corresponder a cualquier tipo de función (agregación, suma, máximo, etc.)[4].

$$R(k2\,,\,list(v2\,)) \rightarrow list(v3)$$

Arquitectura Map Reduce

En la Figura 2 se muestra la arquitectura MapReduce, al igual que con HDFS, MapReduce cuenta con una arquitectura maestro-esclavo y con dos tipos de servicios que conforman su arquitectura:

- Servicio JobTracker: el encargado de gestionar, monitorizar y distribuir la carga de los trabajos MapReduce.
- Servicio TaskTracker: se ejecuta en un nodo con un servicio DataNode de HDFS. Recibe las órdenes del JobTracker y se encarga de realizar el proceso sobre el bloque de datos que contenga.

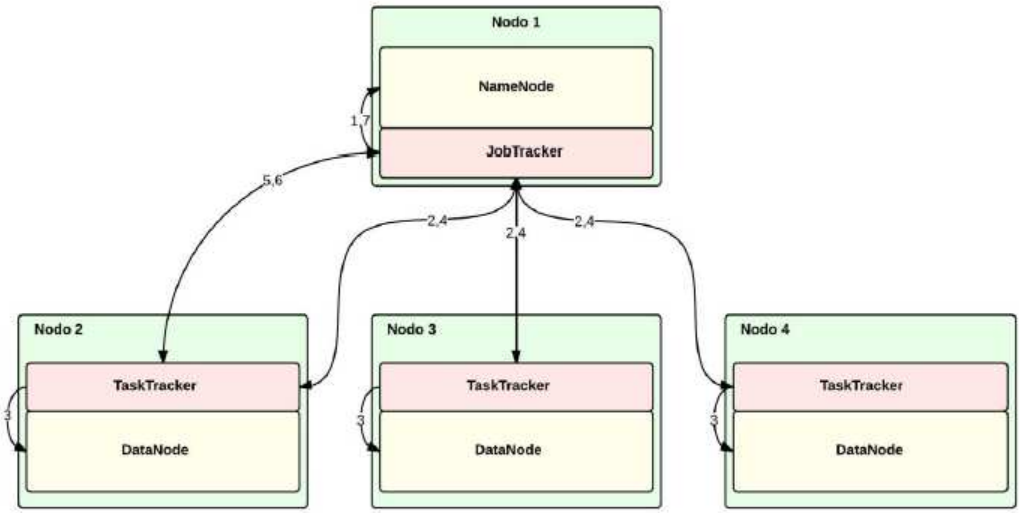


Fig 2. Arquitectura Map Reduce [4]

POWER VIEW

Power View es parte de las herramientas de Power BI que ayudan al análisis e interpretación para casos de inteligencia de negocios. Esta característica viene disponible a partir de la versión de Microsoft Office Excel 2013. Power View es una herramienta usada para la visualización interactiva de diagramas, gráficos, cuadros, entre otros. Los datos representados de esta manera pueden ser analizados con mayor facilidad para la toma de decisiones. En la figura 3 se muestra un informe gráfico generado con esta herramienta.

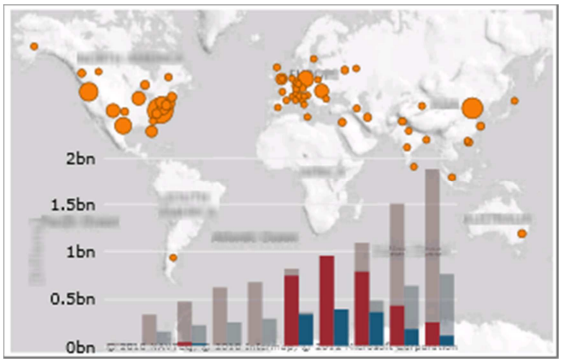


Fig 3. Informe Power View [14]

3 RESULTADOS Y DISCUSIÓN

Microsoft Azure cuenta con un módulo HortonWorks Hadoop con una máquina Virtual preinstalada que permite activar los servicios de Hadoop. Esta fase de activación consiste en iniciar el servidor Apache Ambari el cual cuenta con una interfaz gráfica, intuitiva y fácil de usar que permite gestionar Hadoop y sus componentes como se puede observar en la figura 4 y entre los que se encuentran:

- La Base de Datos Hive
- El Sistema de archivos Distribuido
- PIG
- MapReduce

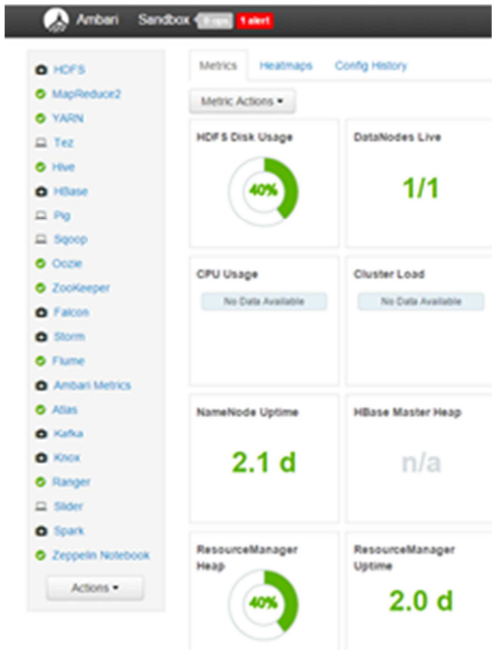


Fig 4. Características de Hadoop en la Interfaz gráfica de Apache Ambari [5]

HDFS es un componente central de Hadoop que permite el almacenamiento homogéneo de la información basándose en una arquitectura Maestro- Esclavo la cual podemos ver en la Figura 5. La función de este componente es separar el fichero de datos en pequeños bloques de 64 MB [15]. Para la selección y carga de datos en HDFS se selecciona el dataset de HortonWorks [5] que contiene la información que se desea analizar. En la presente investigación se trata del dataset que contiene la información de California- EEUU relacionada a rutas vehiculares GPS, información de conductores, información de los camiones e incidentes en las carreteras.

El nodo Maestro (Namenode) gestiona los ficheros y los metadatos o bloques. Los nodos Esclavos (Data node) se encargan de almacenar y recuperar los bloques, además de generar un Clúster de replicación de estos, brindando tolerancia

a fallos. El Dataset utilizado en la presente investigación tiene un tamaño de 100 MB. Al cargarlo en el HDFS convierte el fichero principal en 2 bloques de 64 MB cada uno, con esto se consigue minimizar el tiempo de respuesta por búsquedas.

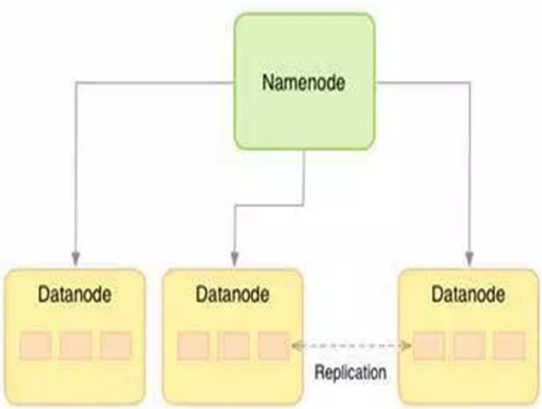


Fig 5. Arquitectura de HDFS [15]

MANIPULACIÓN DE LOS DATOS CON HIVE

Hive es la base de datos propia de Hadoop, en ella es posible realizar las consultas y el análisis necesario de los Dataset seleccionados para el estudio. Dentro del componente Hive, en el editor de sentencias SQL fueron creadas 4 tablas:

- 2 tablas para organizar los datos en el formato csv.
- 2 tablas para el almacenamiento ORC.

Luego de creadas las tablas en Hive es posible cargar los datos de dos formas diferentes como se observa en la Figura 6. La primera forma es moviendo el archivo csv al directorio asociado con la tabla. En este método se usa Load Sample Data. Esta opcion permitirá cargar los datos a la tabla Hive en su formato de archivo csv. El resultado se podrá visualizar de manera gráfica para verificar que los datos se han cargado correctamente.

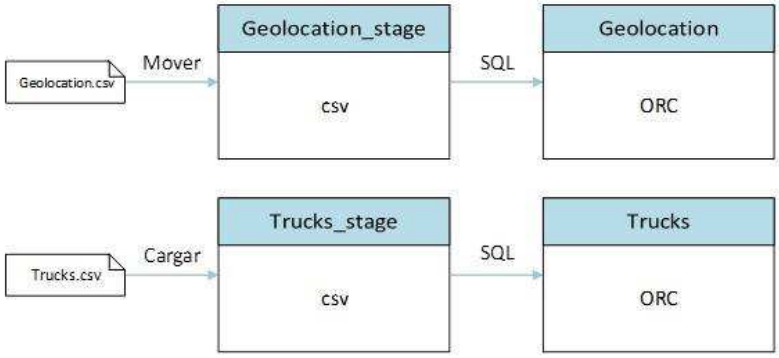


Fig 6. Formas de crear y cargar tablas en Hive

La segunda forma es abriendo una nueva hoja de trabajo y escribiendo la línea de comandos:

LOAD DATA INPATH '/tmp/admin/data/trucks.csv OVERWRITE INTO TABLE trcks_stage;

CREACIÓN DE TABLAS ORC EN HIVE

ORC (Optimized Row Columnar) es un formato de archivo muy eficiente para almacenar datos en Hive, la principal razón consiste en que crea particiones de la tabla siendo beneficioso para las consultas que solo necesitan examinar algunas particiones de la tabla y no todo el volumen de información. El proceso de crear una tabla ORC se realiza desde el editor de secuencias SQL de Hive en la cual se ejecuta la siguiente sentencia:

*CREATE TABLE geolocation STORED AS ORC AS SELECT * FROM geolocation_stage;*

Este código crea una nueva tabla tipo ORC llamada geolocation y carga en ella todos los datos que encuentre en la tabla geolocation_stage; como se muestra en la figura 7. Luego se actualiza el explorador de la base de datos y se visualiza la nueva tabla creada correctamente.

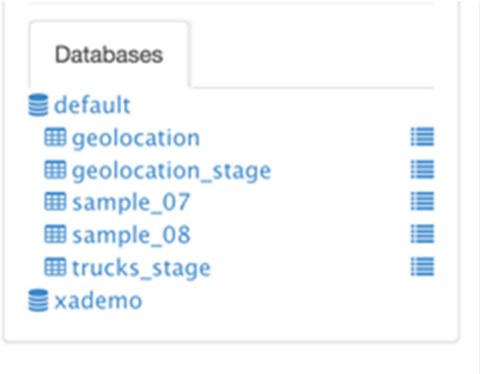


Fig 7. Visualización de la tabla ORC geolocation

ANÁLISIS DE LOS DATOS

Luego de crear las tablas en la base de datos y cargada la información respectiva se procede a analizar la estructura y el contenido de las mismas, para conocer que tablas contienen los campos fundamentales para el proceso de análisis de información. Existen 2 pasos para examinar el contenido de información en las tablas:

- Mediante sentencia SQL directamente en el HIVE.
- Accediendo al diseñador de la base de datos.

En la tabla geolocation_stage se encuentran los campos asociados al código del vehículo, la velocidad a la que viaja y ubicación. Esta tabla permitirá visualizar y trazar una ruta correcta entre dos puntos. Estos datos son utilizados para realizar el análisis en la presente investigación, como por ejemplo análisis de trayectorias GPS vehiculares. Para iniciar el análisis se utilizó el APACHE PIG para calcular los factores de riesgo asociado a cada vehículo que se encuentra en la base de datos. Como condición inicial es necesario crear un script de análisis de riesgo, el cual debe utilizar las sentencias cargadas por defecto en el APACHE PIG y luego proceder a cambiar las variables de interés para la investigación. El script es creado utilizando la sentencia siguiente:

a = LOAD 'geolocation' using org.apache.hive.hcatalog.pig.HCatLoader();

Una vez creado el script se filtrarán los datos para ver de manera más ordenada la información que se desea obtener como parte del análisis. Los factores de riesgo en este caso serán “anormal” o “normal” dependiendo del caso de cada conductor obtenido utilizando un complemento al final del script, el cual quedaría de la siguiente manera:

a = LOAD 'geolocation' using org.apache.hive.hcatalog.pig.HCatLoader();

b = filter a by event != 'normal';

c = foreach b generate driverid, event, (int) '1' as occurance;

d = group c by driverid;

e = foreach d generate group as driverid, SUM(c.occurance) as t_occ;

g = LOAD 'drivermileage' using org.apache.hive.hcatalog.pig.HCatLoader();

h = join e by driverid, g by driverid;

final_data = foreach h generate $0 as driverid, $1 as events, $3 as totmiles, (float) $3/$1 as riskfactor;

store final_data into 'riskfactor' using org.apache.hive.hcatalog.pig.HCatStorer();

Una vez realizadas las configuraciones y creado el script se realiza un test al mismo con ayuda del editor en la opción “execute”. El test automáticamente realizará el análisis de los datos cargados y si no se presenta ningún inconveniente en el proceso se podrá presentar el resultado del análisis con éxito con se observa en la Figura 8.

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns... previous next

riskfactor.driverid	riskfactor.events	riskfactor.totmiles	riskfactor.riskfactor
A1	80	628507	7856.33740234375
A2	80	664543	8306.787109375
A3	80	639584	7994.7998046875
A4	80	663289	8291.1123046875
A5	80	676574	8457.1748046875
A6	80	648479	8105.9873046875
A7	80	653787	8172.33740234375
A8	80	653991	8174.8876953125
A9	80	665456	8318.2001953125
A10	80	675377	8442.212890625
A11	80	652452	8155.64990234375

Fig 8. Resultado del script de análisis de riesgo.

PRESENTACIÓN DE LOS DATOS

Una vez analizados y procesados lo datos se puede visualizar los resultados utilizando diferentes herramientas. En el caso de la presente investigación se utilizó Microsoft Excel Professional Plus 2013 y Power View para generar varios gráficos que permitirán entender mejor el resultado del estudio. Para presentar los datos en la Herramienta Microsoft Office Excel se debe establecer un enlace con Hadoop. Este enlace se establece accediendo o exportando los datos para luego visualizarlos. Para acceder a los datos se crea una hoja en blanco de Excel. En la barra de herramientas *datos* se selecciona *obtener datos externos* y se exporta los datos de Hadoop hacia nuestra hoja de Excel. Se elige el origen de datos, que en este caso será Hadoop y se selecciona la tabla que se haya procesado en el Hadoop. Excel enviará una solicitud a Hadoop y finalmente mostrará los datos completos de la tabla seleccionada como se observa en la Figura 9.

	A	B	C
1	truckid	avgmpg	
2	A1	4.785822711	
3	A10	5.401717664	
4	A100	4.939038953	
5	A11	5.502368693	
6	A12	4.686163839	
7	A13	5.82054548	
8	A14	4.939907303	
9	A15	5.009636162	
10	A16	4.933367405	
11	A17	4.902493925	
12	A18	6.295206724	
13	A19	5.0493527	
14	A2	5.795686564	
15	A20	4.4346379	

Fig 9. Datos extraídos de Hadoop

VISUALIZACIÓN DE LOS DATOS

Una vez cargados e importados los datos de manera general en Excel, son procesados para visualizarlos en un mapa. Para ello se recurre a la Función Power View, que permite presentar un informe gráfico de los datos procesados. Para crear un informe Power View se selecciona la opción insertar en la barra de Herramientas y luego la función Power View Reports. Se crea una hoja Power View para procesar y visualizar los datos. La visualización de forma gráfica seleccionando la opción diseño/gráfico de columnas muestra un reporte como el que se puede apreciar en la Figura 10.

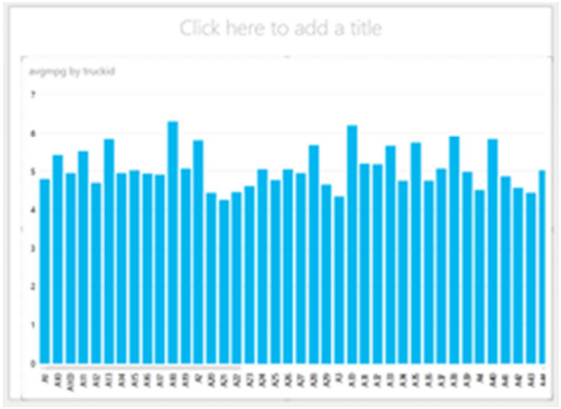


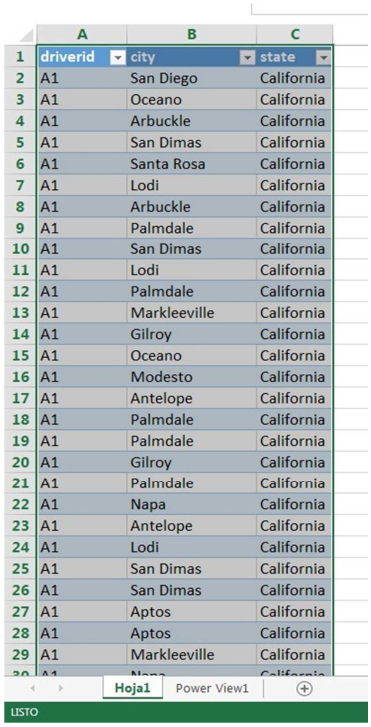
Fig 10. Reporte Gráfico en Power View

Debido a que la presente investigación consiste en la geo localización de diferentes puntos es de vital importancia visualizar al mapa generado. De esta manera es posible tener una mejor visualización y apreciación de los datos localizados en un mapa. Para ello se realiza un mapa de hechos utilizando los datos almacenados en la tabla Geolocalización (Esta tabla describe los datos a procesar para la elaboración del mapa). Los pasos para crear el mapa de hechos se definen a continuación:

- Seleccionar el contenido de las columnas a utilizar. En este caso se toman los datos asociados a DriverID, city y estate; mediante la sentencia: select driverid, city, state from geoleocation;
- Capturar los datos en un nueva tabla mediante la siguiente sentencia: CREATE TABLE events STORED AS ORC AS select driverid, city, state from geoleocation;
- Al ejecutar esta consulta los eventos en la tabla se crean, otra opción es importando los datos desde Excel como se observó anteriormente. En cualquier caso se obtiene el resultado que se muestra en la Figura 11
- Finalmente la información extraída debe ser pasada a un informe Power View y se selecciona la opción *mapa*.

INTERPRETACIÓN DE LOS RESULTADOS

Los datos obtenidos de la Imagen 10, muestran el promedio de millas recorridas por galón de combustible que consume cada camión. Esto ayudará a tomar decisiones en cuanto al correcto funcionamiento de los motores de los distintos camiones. Otro ejemplo podría ser la selección de rutas más eficaces para el ahorro de combustible. Los Datos obtenidos en la Figura 12 presentan las ciudades donde han estado presente los camiones, esto permite observar que los camiones no se desvíen de su recorrido y están en los lugares donde se les asignó. Para un jefe de departamento, por ejemplo, este informe es útil para el control de las unidades vehiculares.



	A	B	C
1	driverid	city	state
2	A1	San Diego	California
3	A1	Oceano	California
4	A1	Arbuckle	California
5	A1	San Dimas	California
6	A1	Santa Rosa	California
7	A1	Lodi	California
8	A1	Arbuckle	California
9	A1	Palmdale	California
10	A1	San Dimas	California
11	A1	Lodi	California
12	A1	Palmdale	California
13	A1	Markleeville	California
14	A1	Gilroy	California
15	A1	Oceano	California
16	A1	Modesto	California
17	A1	Antelope	California
18	A1	Palmdale	California
19	A1	Palmdale	California
20	A1	Gilroy	California
21	A1	Palmdale	California
22	A1	Napa	California
23	A1	Antelope	California
24	A1	Lodi	California
25	A1	San Dimas	California
26	A1	San Dimas	California
27	A1	Aptos	California
28	A1	Aptos	California
29	A1	Markleeville	California

Hoja1 Power View1

LISTO

Fig 11. Datos para el Mapa de Hechos

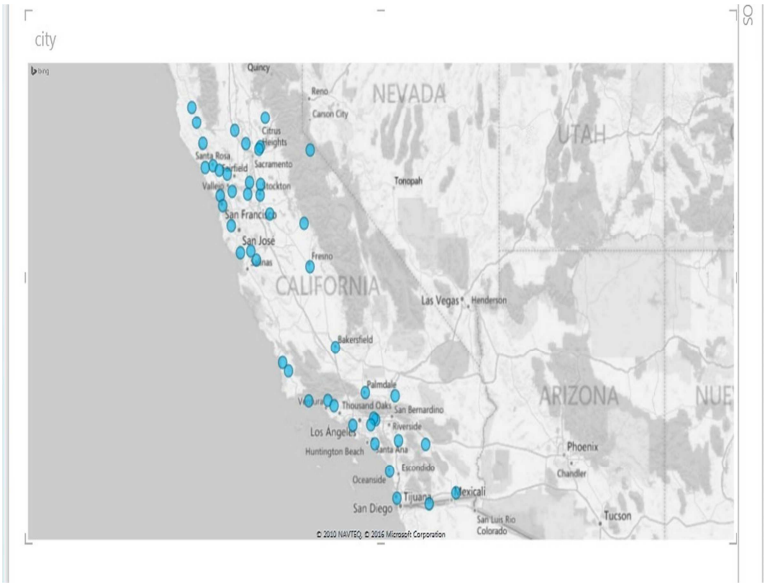


Fig 12. Mapa de Hechos

4 CONCLUSIONES

En la presente investigación se analizaron 8112 registros de los cuales 8012 se procesaron para formar la tabla *events*, lo que permitió un análisis profundo de los datos y la generación de mapas con abundante información. El procesamiento de distintos tipos de información asistió a la generación de informes utilizados en la toma de decisiones. La conversión a formato ORC facilitó la optimización de filas y columnas en la base de datos. La utilización del framework Hadoop ha permitido manipular y generar informes de grandes cantidades de datos, que con otro tipo de herramientas no habría sido posible. Como trabajo futuro se propone la utilización del framework Hadoop en diferentes proyectos que se están iniciando en la organización para el manejo y evaluación de grandes cantidades de datos. Adicionalmente se prevé generar rutas GPS de la ciudad de Guayaquil para asistir al desarrollo científico de esta rama en el país.

BIBLIOGRAFÍA

[1] WINSHUTTLE. (2013). Obtenido de http://www.winshuttle.es/big-data-historia-cronologica/
[2] Fundación Innovación Bankinter. (2014). Obtenido de : http://www.fundacionseres.org/Lists/Informes/Attachments/959/150528%20Big%20Data%20ES%20Completo%202015.pdf
[3] Ticout Outsourcing Center. (2014). Obtenido de Explotemos la Tecnología: http://www.ticout.com/blog/2013/04/02/introduccion-a-hadoop-y-su-ecosistema/
[4] Morros, R. S. (Noviembre de 2013). BIG DATA- ANÁLISIS DE HERRAMIENTAS Y SOLUCIONES. Everis – Facultat d'Informàtica de Barcelona – UPC.
[5] HortonWorks. (2015). Obtenido de http://hortonworks.com/
[6] Apache Hive. Apache Hive. [En línea] The Apache Software Foundation. [Consultado el: 5 de Abril de 2013.] https://cwiki.apache.org/confluence/display/Hive/Home.
[7] Chen, Charles. Diseño. Apache Hive. [En línea] The Apache Software Foundation, 21 de Julio de 2011. [Consultado el: 5 de Abril de 2013.] https://cwiki.apache.org/confluence/display/Hive/Design.
[8] tutorialspoint. (2013). Obtenido de http://www.tutorialspoint.com/es/hadoop/hadoop_hdfs_overview.htm
[9] Aguinaga, J. (3 de Mayo de 2014). Tech Net Blogs. Obtenido de : http://blogs.technet.com/b/jorge_aguinaga/archive/2014/05/03/191-qu-233-es-power-view.aspx
[10] Pazmiño, M. A., Torralbo, J. A., & Casas, D. L. (2015). Framework para la gestión, el almacenamiento y preparación de grandes volúmenes de datos. CEF, 17.
[11] Welcome to Apache Pig! Apache Pig. [En línea] The Apache Software Foundation, 22 de Octubre de 2013. [Consultado el: 16 de Abril de 2013.] http://pig.apache.org/
[12] Microsoft Azure. (2014). Obtenido de Microsoft Azure: https://azure.microsoft.com/es-es/documentation/articles/machine-learning-data-science-move-hive-tables/#submit
[13] Gomez, J. I. (2014). Departamento de Economía. Obtenido de Universidadd de La Laguna: http://www.jggomez.eu/K%20Informatica/3%20Excel/03%20Mis%20Temas/C%20Dashboard/Power%20BI.pdf
[14] Office Support. (2014). Obtenido de https://support.office.com/es-es/article/Power-View-explorar-visualizar-y-presentar-los-datos-98268d31-97e2-42aa-a52b-a68cf460472e
[15] Garcia, L. M. (23 de Julio de 2013). Un Poco de Java. Obtenido de : https://unpocodejava.wordpress.com/2013/07/24/que-es-hdfs/