

Facebook y su correlación con Big Data y Hive

Freddy Tapia León, MSc.

Universidad de las Fuerzas Armadas – ESPE, Ecuador, fntapia@espe.edu.ec

Abstract– We live in an era of “Big Data” produced by a wide range of applications, but facing the problem of scalability and high speed access. This paper focuses its analysis in this type of situation, and provides special support for batch oriented analytics applications. Facebook Data Infrastructure supports a wide range of applications, often this infrastructure is built on open-source technologies such as Apache Hadoop, Hadoop Distributed File System (HDFS), MapReduce and Hive. In this work the main idea is to understand how MapReduce facilitates parallel databases accessibility, include storage-system independence and fine-grain fault tolerance for large data sets. The main contribution is to show future works regarding this area.

Keywords-- Hadoop, Hive, MapReduce, HDFS

I. INTRODUCCIÓN

De acuerdo con IBM [1], cada día se generan alrededor de 2.5 quintillones de bytes de datos (10^{30}), donde el 90% han sido creados en los últimos tres años. A su gestión eficiente se le conoce como Big Data, la cual es una nueva oportunidad para proponer ideas, generar análisis, reutilización de datos y contenidos, lo que optimiza una toma de decisión más adecuada. Facebook actualmente es la más grande red social, por ende es la mayor fuente de información no solo por el número de usuarios que tiene, sino también por los servicios y aplicaciones que ofrece, todo esto es posible gracias a la infraestructura de datos con la que cuenta. Esta investigación analiza como Facebook maneja su información, así como también las tecnologías asociadas, como por ejemplo: Apache Hadoop, HDFS, MapReduce y Hive, las cuales son fundamentales para la consolidación y funcionamiento de esta red social [2].

El resto del artículo ha sido organizado como sigue: La sección II describe el fundamento teórico. La sección III muestra los componentes involucrados en este análisis. La sección IV evidencia los resultados. Finalmente en la sección V se exponen las conclusiones y trabajos futuros.

II. MARCO REFERENCIAL

A. Hadoop Distributed File System (HDFS)

Sistema distribuido, escalable y portátil creado especialmente para trabajar con ficheros de gran tamaño, los cuales son divididos en bloques de un mismo tamaño y distribuidos entre los nodos que forman el clúster de datos.

Un clúster HDFS utiliza dos tipos de nodos: (1) Namenode (JobTracker), de este tipo de nodo hay uno solo por clúster, el cual es responsable de la topología y distribución de procesos entre los demás nodos; (2) Datanodes (TaskTracker), de este tipo de nodos van a existir varios, los cuales realizan

el acceso y almacenamiento de la información (bloques), así como también la recuperación bajo demanda.

B. Hive

Solución de almacenamiento masivo de datos, el cuál integra funcionalidades de búsqueda a través de Apache Hadoop, las cuales son similares a RDBMS (Relational Database Management System), así como también usa especificaciones HiveQL, las cuales son similares a SQL.

Hive destaca la optimización de consultas, debido a su estabilidad y rendimiento, más aún cuando el rendimiento es medido por el tiempo de respuesta y cantidad de procesos ejecutados [3].

C. Hadoop

Provee un modelo confiable, escalable para ambientes distribuidos a gran escala, por medio del procesamiento en paralelo. Este modelo ofrece un almacenamiento local y difusión de información entre los nodos y clústeres definidos, garantizando una alta disponibilidad y detección de fallas, lo que permite entregar servicios altamente confiables [4].

D. MapReduce

Modelo de programación diseñado para administrar grandes cantidades de datos (PetaBytes), el cual es utilizado para dar soporte a la computación paralela, este modelo arroja resultados a ficheros HDFS como archivos del sistema, lo que garantiza una alta disponibilidad y confiabilidad de los datos.

Por lo tanto, la paralelización y procesamiento por lotes aceleraría la ejecución de las consultas, así como la ejecución de programas dentro de grandes clústeres de una manera distribuida, generando interdependencia en el almacenamiento y granularidad fina aplicada [5][7].

III. MATERIALES Y MÉTODOS

La Fig. 1, muestra el flujo de datos utilizado por Facebook, la cual inicia con el paso de los datos (Web Servers), hacia un conjunto de Clústeres Scribe-Hadoop (alrededor de 50 nodos por bloque), estos servidores agregan registros a los datos provenientes de los diferentes servidores web y su posterior transformación en archivos HDFS, para posteriormente asignar prioridades de ejecución por medio del Production Hive-Hadoop (alta prioridad), y Adhoc Hive-Hadoop (baja prioridad). Hive se encarga de sincronizar y verificar las tablas de asignación, de aquí la necesidad que los datos sean cargados primero en el cluster de producción y no

el adhoc (replica), evidenciando así la escalabilidad que maneja MapReduce (tiempos de respuesta), para finalmente interactuar constantemente con el almacenamiento de datos (Federated MySQL), por medio de procesos “scraping” (técnicas de extracción de datos-MetaStore), y así tener datos siempre disponibles [6].

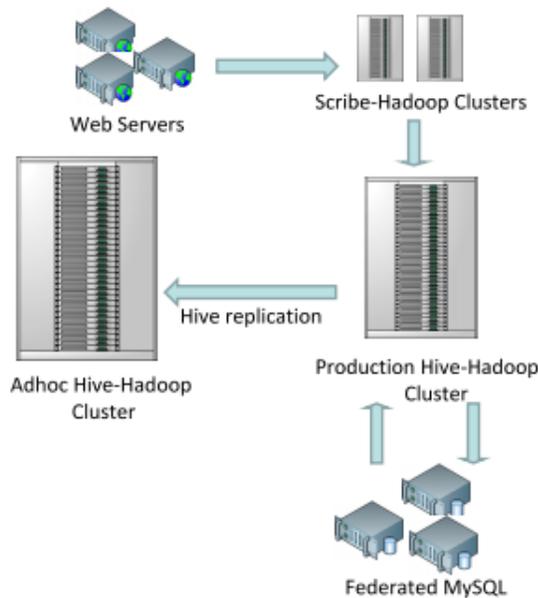


Fig. 1. Flujo de Datos – Facebook

IV. RESULTADOS

Los resultados mas evidenciados son: (1) Tener la capacidad de mantener varias estructuras de datos; (2) Optimización del ancho de banda debido al almacenamiento en disco locales de las maquinas que conforman los clusters, así como también la utilización de datos válidos en lugar de esperar a que los datos sean transferidos por los usuarios; (3) El sistema de almacenamiento MapReduce es independiente y puede procesar datos sin necesidad de estar dentro de la base de datos.

V. CONCLUSIONES

El presente trabajo se basó en una investigación descriptiva, la cual busca comprender los procesos de manejo y análisis de los datos generados por Facebook.

Tal es así que MapReduce y Hive, son usados para procesar búsquedas en bases de datos a gran escala (big data), por medio de escaneos secuenciales y agregaciones.

Existen muchas bondades del modelo “map” y “reduce”, así como también existen nuevas propuestas en ejecución, tal es el caso de MapReduce2, el cual se orienta a mejorar algunas funciones de su antecesor (JobTracker).

La granularidad fina, es una propiedad que los sistemas distribuidos y paralelos obtienen producto del procesamiento que desarrollan.

La relación que existe entre las Bases de Datos NoSQL y SQL, son similares a las aplicadas por HiveQL, volviéndola amigable y de fácil adaptación.

El modelo MapReduce permite realizar búsquedas y minería de datos a través de aprendizajes elaborados por sistemas inteligentes, así como también incorporando principios semánticos para las búsquedas y resultados obtenidos.

Facebook, construye una serie de sistemas y plataformas alternas, con el objetivo de ir evaluándolas y de esta forma ir escalando hacia tecnologías más amigables y de fácil uso.

Como trabajos futuros se pretende realizar pruebas de desempeño sobre MapReduce Merge y ReStore, los cuales incorporan algebra relacional y reutilización de los datos HDFS.

Optimizar el algoritmo de paralelización y mejorar la tolerancia a fallos y velocidad de procesamiento

RECONOCIMIENTO

Esta investigación ha sido posible gracias a la contribución que la Universidad de las Fuerzas Armadas – ESPE, viene ejecutando con todos los docentes investigadores que actualmente están cursando sus estudios doctorales.

REFERENCIAS

- [1] IBM, “Big Data at the Speed of Business”. <http://www-01.ibm.com/software/data/bigdata/>.
- [2] A. Menon, “Big Data @ Facebook”. <http://www.cercs.gatech.edu/mbds12/talks/aravind.pdf>
- [3] A. Gruenheid, E. Omiecinski, and L. Mark, “Query Optimization Using Column Statistics in Hive”, pp. 97-105, 2011
- [4] The Apache Software Foundation. <http://hadoop.apache.org/>
- [5] The Apache Software Foundation, http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
- [6] A. Thusoo, D. Borthakur, R. Murthy, “Data Warehousing and Analytics Infrastructure at”. <http://borthakur.com/ftp/sigmodwarehouse2010.pdf>
- [7] J. Dean, S. Ghemawat, “MapReduce: simplified data processing on large clusters”, pp.107-113, 2008.