Apache Pig en Hadoop sobre Cassandra

Susana B. Chavez[1], Adriana E.Martin[2], Nelson R. Rodríguez[3], María A. Murazzo[4]
Departamento e Instituto de Informática - F.C.E.F. y N. - U.N.S.J.
Complejo Universitario Islas Malvinas, Av. I. de la Roza 590 (O), CP: 5402 Rivadavia, San Juan. Tel:0264 4234129

[1]schavez@iinfo.unsj.edu.ar [2]arianamartinsj@gmail.com [3]nelson@iinfo.unsj.edu.ar
[4]marite@unsj-cuim.edu.ar

Resumen

Los sistemas distribuidos en la web y las tecnologías informáticas distribuidas como cluster y cloud, permiten diseñar un entorno de entidades distribuidas que cooperen para resolver un problema que no puede ser resuelto individualmente. La variedad de estos sistemas pueden incluir servidores de aplicaciones, cloud privados, pequeños centros de datos y cluster para almacenamiento y búsqueda de datos.

Esto explica por qué ha crecido enormemente la habilidad de recolectar y almacenar datos en las últimas décadas, incluso hoy en día, se pude decir que este apetito por los datos no muestra signos de satisfacción.

Los científicos quieren ser capaces de almacenar más datos con el fin de construir mejores modelos matemáticos del mundo. Los vendedores quieren mejores datos para entender los deseos y hábitos de compra de sus clientes. Los analistas financieros quieren entender mejor el funcionamiento de sus mercados. Y todo el mundo quiere mantener todas sus fotografías, videos, correos electrónicos, etc.

En consecuencia, es primordial encontrar la mejor solución para el procesamiento y análisis de esta gran escala de enormes cantidades de datos.

En este sentido, un RDBMS como SQL Server o MySQL es una buena opción si el conjunto de datos de trabajo nunca va a crecer más allá de 40-50GB a lo largo de su vida útil. Incluso no necesitan ser distribuidos ya que pueden ser procesados en la memoria de una sola máquina.

Sin embargo, si se construye una aplicación que tiene un conjunto de datos que crece rápidamente y ráfagas de cargas impredecibles, será necesario optar por una solución que sacrifique cierta velocidad o consistencia en pos de poder distribuirse y así procesar el gran volumen de datos.

En los últimos años han surgido las bases de datos NoSQL que rompen una o más de las reglas de los sistemas de bases de datos relacionales. No esperan que los datos sean normalizados. En su lugar, los datos a los que accede una aplicación viven en una gran tabla, de modo que pocos o ningún *joins* son necesarios. Estos sistemas están diseñados para administrar terabytes de datos.

A esto, se suma el desarrollo de muchos sistemas alternativos de procesamiento de datos como *Apache Hadoop*. Este proyecto ha impulsado el desarrollo de lenguajes existentes y la construcción de nuevas herramientas como *Apache Pig*. Esta herramienta proporciona un

mayor nivel de abstracción para los usuarios de datos, dando acceso a la flexibilidad y potencia de Hadoop sin necesidad de tener que escribir extensas aplicaciones de procesamiento de datos en código Java de bajo nivel.
Las bases de datos NoSql que se han integrado con Pig incluyen HBase, Accumulo y Cassandra.
En este trabajo se propone realizar pruebas experimentales con Apache Pig sobre Apache Hadoop y como motor NoSql se elige Cassandra, ya que coincide muy bien con la naturaleza distribuida de Hadoop, para ejecutar consultas sobre datos que abarcan múltiples nodos.

Palabras Claves: Apache Pig, Hadoop, Nosql ,Computación distribuida, Cloud Computing.

Contexto

El presente trabajo se enmarca dentro del proyecto de investigación: **Evaluación de arquitecturas distribuidas de commodity basadas en software libre**, el cual tiene como unidades ejecutoras al Departamento e Instituto de Informática de la FCEFyN de la UNSJ.

Introducción

Los sistemas de bases de datos NoSql crecieron con las principales redes sociales, como Google, Amazon, Twitter y Facebook, debido a que debieron enfrentarse a nuevos desafíos con el tratamiento de datos. Con el crecimiento de la web en tiempo real existía una necesidad de proporcionar información procesada a partir de grandes volúmenes de datos que tenían estructuras horizontales más o menos similares.

En ese sentido, las bases de datos NoSql están altamente optimizadas para las operaciones de recuperar y agregar, y normalmente no ofrecen mucho más que la funcionalidad de almacenar los registros. La pérdida de flexibilidad en tiempo de ejecución, comparado con los sistemas SQL clásicos, se ve compensada por la significativa ganancia en escalabilidad y rendimiento cuando se trata con ciertos modelos de datos [1].
Apache Cassandra es una base de datos NoSql distribuida y de código abierto, cuya principal característica es que fusiona Dynamo, de Amazon con BigTabla, de Google, siendo ambas implementaciones de código cerrado. Está basada en un modelo de almacenamiento de «clave-valor», escrita en Java. Permite almacenar grandes volúmenes de datos en forma distribuida. Su objetivo principal es la disponibilidad y la escalabilidad lineal [2][3].

"Apache Cassandra se ha convertido en una de las bases de datos NoSQL más utilizados del mundo y sirve como columna vertebral de algunas aplicaciones muy populares hoy en día". Un ejemplo de esto, es Twitter que lo usa en su plataforma. Su objetivo principal es la escalabilidad lineal y la disponibilidad. Permite el uso de Hadoop para implementar MapReduce, ya que Hadoop puede trabajar directamente con cualquier sistema de archivos distribuido [4].

Apache Hadoop, surge como una alternativa para el procesamiento de estos datos masivos [5]. Hadoop es un proyecto de código abierto iniciado por Doug Cutting. Durante los últimos años, Yahoo! y varias otras compañías web han impulsado el desarrollo de Hadoop, basado en artículos publicados por Google que describen cómo enfrentaban al reto de almacenar y procesar las

enormes cantidades de datos que eran coleccionadas. Hadoop se instala en un grupo de máquinas y proporciona un medio para unir el almacenamiento y el procesamiento en un clúster.

El desarrollo Hadoop ha impulsado el desarrollo de herramientas y lenguajes existentes y la construcción de nuevas herramientas como Apache Pig [6].

Pig proporciona un motor para ejecutar flujos de datos en paralelo en Apache Hadoop. Incluye un lenguaje, Pig Latin, para expresar estos flujos de datos. Pig Latin incluye operadores para muchas de las operaciones de datos tradicionales (join, sort, filter, etc.), así como proporcionar a los usuarios la capacidad de desarrollar sus propias funciones para leer, procesar y escribir datos.

Pig Latin es un lenguaje de flujo de datos. Esto significa que permite a los usuarios describir cómo los datos de una o más entradas deben ser leídos, procesados y luego almacenados en una o más salidas en paralelo. Estos flujos de datos pueden ser simples flujos lineales o flujos de trabajo complejos que incluyen puntos donde se unen múltiples entradas y donde los datos se dividen en múltiples flujos para ser procesados por diferentes operadores. No hay instrucciones if o para bucles en Pig Latin. Esto se debe a que los lenguajes de programación tradicionales y orientados a objetos describen el flujo de control y el flujo de datos es un efecto secundario del programa. En cambio, Pig Latin se centra en el flujo de datos.

Pig corre en Hadoop. Utiliza el Sistema de Archivos Distribuidos Hadoop (HDFS) y el sistema de gestión de recursos de Hadoop (YARN, a partir de Hadoop 2). HDFS es un sistema de archivos distribuido que almacena archivos en todos los nodos de un clúster Hadoop. Se encarga de romper los archivos en bloques grandes y distribuirlos a través de diferentes máquinas, incluyendo la realización de copias múltiples de cada bloque para que si una máquina falla, no se pierden datos. Presenta una interfaz similar a POSIX a los usuarios. De forma predeterminada, Pig lee los archivos de entrada de HDFS, utiliza HDFS para almacenar datos intermedios entre los trabajos de MapReduce y escribe su salida en HDFS [7].

Líneas de Investigación y desarrollo

De acuerdo a la literatura consultada, los casos de usos de Pig latino se clasifican en tres grupos: los procesos tradicionales de extracción, transformación y carga (ETL), la investigación de los datos brutos (raw data) y el procesamiento iterativo.

El objetivo de esta línea de investigación dentro del proyecto marco, es construir modelos de predicción de comportamiento. Apache Pig se presenta como una herramienta optima para explorar todas las interacciones del usuario con un sitio web y para luego dividir a los usuarios en varios segmentos. En este caso se tomara el sitio de la UNSJ como primera experiencia.

Paso seguido, se deberá construir un modelo por cada segmento, este modelo predecirá como los miembros de ese segmento responderán a los tipos de anuncios o noticias. De esta forma, el sitio web podrá mostrar anuncios con mayor probabilidad de obtener un clic u ofrecer noticias que tengan más probabilidades de involucrar a los usuarios y que vuelvan al sitio. Por qué Pig Latin?:

- Porque puede operar en situaciones donde el esquema es desconocido, incompleto o inconsistente, además puede manejar fácilmente los datos anidados y

- Porque es un proyecto Apache open source. Esto significa que se puede descargar como fuente o binario, utilizarlo para trabajo experimental, contribuir a él y, bajo los términos de la Licencia Apache, usar sus productos y adaptarlos como mejor convenga.

Resultados Obtenidos

A pesar de que esta línea de investigación ha sido presentada para los años 2016 y 2017. Durante los últimos cinco años se trabajó en proyectos sobre Cloud Computing y en particular durante los últimos dos años sobre Cloud híbridos. La experiencia sobre los Cloud privados, junto con líneas de investigación anteriores, impulsó esta línea de investigación. El grupo ha realizado nueve publicaciones en el área durante el último año: tres trabajos en el WICC 2015-2016, un trabajo en el CACIC 2015, dos trabajos en las Jornadas de Cloud Computing, además se realizaron tres publicaciones en revistas científicas. Se han aprobado tres tesinas de grado y un trabajo de especialización.

Objetivo

En este proyecto se enfocaran las investigaciones en los sistemas de cómputo distribuidos, los cuales permiten realizar de manera más eficiente tareas de computación de alta prestaciones basadas en el paradigma de memoria distribuida. Ejemplos de Arquitecturas que soportan este tipo de sistemas distribuidos son los cluster y el cloud computing.

En particular este trabajo tiene como objetivo instalar una base de datos Nosql, en particular Cassandra, sobre un cluster montado como servicio (CaaS), para luego utilizar Apache Pig sobre Hadoop para realizar el procesamiento de los datos.

Formación de Recursos humanos

El equipo de trabajo está compuesto por seis docentes-investigadores y cuatro alumnos.
Se están realizando cuatro tesinas de licenciatura una sobre evaluación de algoritmos de algebra lineal sobre arquitecturas diversas, otra sobre Cloud Computing Privado, otra sobre dispositivos de juegos aplicados a salud y otra sobre SOA aplicada a Cloud. Se espera realizar también una tesis de maestría sobre Metodologías de desarrollo aplicadas a SaaS, otra sobre bases de datos NoSQL y otra sobre algoritmos de Cómputo Intensivo para Big Data y su implementación en Clouds. Además aumentar el número de publicaciones. Por otro lado también se prevé la divulgación de varios temas investigados por medio de cursos de postgrado y actualización o publicaciones de divulgación.

Referencias

[1] http://wikipedia.org/wiki/NoSQL

[2] http://cassandra.apache.org

[3] C.Y. Kan, "**Cassandra Data Modeling and Analysis**" Copyright © 2014 - Packt Publishing.

[4] www.siliconweeks.com

[5] Tom White, "**Hadoop: The Definitive Guide**" Copyright © 2015 - O′Reilly

[6] http://pig.apache.org/

[7] Alan Gates y Daniel Dai, "**Programming Pig**" Copyright © 2016 - O′Reilly Media