PRACTICA

ARQUITECTURA BIG DATA

Daniel Alvaro Parcio

INDICE

1 Contexto

Una vez finalizado el modulo de Big Data Architect debemos llevara a cabo la practica en la que demostrar el conocimiento adquirido en el mismo.

Para ello, expondremos un caso de uso real, para el que será necesario definir una arquitectura Biga Data, usando como punto de partida el dataset de Airbnb.

Si bien definir el Data Lake, es solo una parte de la Arquitectura Big Data, en un proyecto de estas características y con el nivel que tenemos, se convierte en parte central del documento de especificaciones para dicha Arquitectura. Desde esta perspectiva el presente documento seguirá la guía de diseño de un Data Lake: Estrategia, Arquitectura, Operación y Desarrollo.

2 Introducción

Antes de comenzar cualquier proyecto Big Data se hace fundamental responder a estas dos preguntas (realmente la primera es doble):

1. **¿Qué quiero conseguir y qué información quiero utilizar?**
2. **¿Justifica el punto 1 montar una arquitectura Big Data?**

El punto 1, se refiere a que antes de empezar cualquier proyecto debemos definir un objetivo, una estrategia. Diseñar e implementar una arquitectura Big Data es costoso en tiempo y recursos, por tanto, no tiene sentido empezar a montar dicha arquitectura sin tener claro que es lo que queremos obtener de ella y por tanto si es realmente necesaria.

Una vez que tenemos mas o menos claro el primer punto debemos ser capaces de justificar la necesidad de montar un entorno Big Data. Hadoop, como base de cualquier proyecto Big Data aporta la capacidad de procesamiento en paralelo, así como la capacidad de almacenar cantidades enormes de información repartiéndola entre los distintos nodos del clúster que despleguemos. Sin embargo, y como ya hemos apuntado anteriormente, diseñar e implementar una arquitectura Big Data implica ciertos costes, que es necesario justificar frente a una arquitectura o DataWarehouse tradicional (que casi con toda seguridad ya estará implementado en la empresa)

En nuestro caso, a pesar de que el objetivo inicial puede parecer poco ambicioso para justificar el montar un entorno Big Data*, lo hacemos así con vistas a la futura escalabilidad del proyecto, incorporando, por ejemplo, información de redes sociales (trypadvisor, twitter) e incluso de redes IoT (conexiones con gadgets de domótica instalados en los pisos de Airbnb).

*** Aunque ya solo el echo de tener que guardar el histórico de la información de todas las viviendas del dataset de Airbnb ya justificaría el uso de Hive y de la plataforma de Big Data.**

3 Estrategia

3.1 Objetivo

Antes de definir el objetivo de mi proyecto me gustaría establecer un par de conceptos para tener claro de que estamos hablando.

La plataforma Airbnb tiene dos clases de usuarios:

- Clientes: Son las personas que alquilan un alojamiento por un tiempo determinado.
- Anfitriones: Son las personas que ponen a disposición de la plataforma las viviendas que alquilan los clientes.

Dicho esto, **el objetivo que se persigue con este proyecto de Big Data** es ofrecer un servicio a la propia empresa Airbnb que le permita visualizar las bajas de Anfitriones de su plataforma y analizar los posibles motivos de dichas bajas. Inicialmente se hará solo de las viviendas en Madrid, pero la plataforma se diseñará de manera que sea fácil de escalar para absorber todas las localidades en las que estemos interesados

Este análisis permitirá sacar conclusiones y proceder de manera proactiva para evitar que un Anfitrión se dé de baja, ofreciéndole por ejemplo publicidad gratuita o consejos sobre como mejorar su experiencia en la plataforma.

El servicio ofrecido a Airbnb consistirá en un informe quincenal en el que se reflejaran los Anfitriones dados de baja en la plataforma y las posibles causas de la misma.

Para ello inicialmente se usarán los datos del Dataset de Airbnb: “airbnb-listings.csv” y los datos extraídos por crawling/scraping de la web de milanauncios.com, para extraer información de si por ejemplo algún piso de la plataforma está puesto a la venta, así como el precio por el que se vende, etc.

De cara al futuro y con una visión mucho mas ambiciosa de este proyecto que la inicialmente propuesta, se pueden considerar las siguientes mejoras o ampliaciones:

- Tener en cuenta comentarios en redes sociales (trypadvisor, Twitter) sobre los apartamentos de los Anfitriones e incorporarlo al histórico de información sobre las viviendas.
- Incorporar información sobre “experiencia de usuario” de los clientes a través de dispositivos domóticos de IoT (horas de uso de la casa, uso del aire acondicionado, uso de internet, consumo eléctrico, horarios de entrada/salida). Evidentemente cualquier información usada debería cumplir las normativas vigentes sobre GDPR.
- Ofrecer a Airbnb, además de los informes quincenales, comentados anteriormente, una plataforma con información en tiempo real sobre los

Anfitriones en peligro de darse de baja en fechas próximas. Dicha plataforma permitiría actuar a Airbnb de forma proactiva, en lugar de reactiva como es el caso de los informes.

3.2 Datos

3.2.1 Source

La información que alimenta nuestra plataforma Big Date se extrae fundamentalmente de dos fuentes:

- **DataSet Airbnb**: Se utilizará inicialmente para cargar el histórico de viviendas en HIVE y se actualizara diariamente para ver que pisos han abandonado la plataforma. Si dicho DataSet no se actualiza con la suficiente periodicidad se planteará sacar esa información mediante un crawling de la propia pagina de Airbnb.
- **Web de milanuncios.com**: Mediante el crawling de la misma podremos sacar información sobre pisos de la plataforma que están en venta, precio de venta de los mismos.

La relación entre ambos sources se hará inicialmente por la dirección de los mismos. Se que no un mecanismo demasiado fiable, pero como punto de partida me parece razonable. A medida que avance el proyecto se analizara otros campos que relacionen de manera mas fiable ambos orígenes de datos.

3.2.2 Salida

Se generarán de manera quincenal dos informes:

- Informe Básico: CSV con los Anfitriones que se han dado de baja
- Informe Plus: Cuadro de mandos interactivo realizado con Tableau que mostrara además de las bajas las posibles causas de las mismas.

3.2.3 Tratamiento de datos

Como el dataset de Airbnb será lo que usemos para cargar la BD Hive, lo primero que haremos será filtrar y limpiarlo. Para ello usaremos una herramienta de Data Cleaning llamada Trifacta y que ya vimos en un modulo anterior.

Lo que haremos será:

- Quedarnos solo con los apartamentos de la ciudad de Madrid
- Eliminamos todos los campos que no aportan información a la generación del informe.
- Limpiamos los campos relacionados con la dirección del inmueble.

Respecto a la pagina de milanuncios.com llevaremos a cabo un crawling diario (usando la herramienta Scrappy de Python) para ver posibles pisos de los Anfitriones en ventas y añadir esa información a nuestra BD en HIVE.

4 Arquitectura

4.1 En la nube todo es mas fácil

La opción elegida para montar la infraestructura de Big Data es Google Cloud, las razones a continuación:

- Solución escalable de manera prácticamente inmediata
- Permite crear un clúster de Hadoop con Hive ya instalado en cuestión de minutos (Google DataProc).
- Permite la gestión del Firewall, accesos, puerto de manera sencilla y centralizada.
- Ofrece un segmento de Google Storage que facilita sobre manera el proceso de Staging (es tan sencillo como apretar el botón subir fichero), así como un sitio donde guardar los informes generados.
- En el modulo hemos hecho mucho hincapié en su funcionamiento y creo que ya tengo una base mínima para gestionarlo.
- Airbnb es una multinacional se puede permitir montar un buen Cloud en Google sin necesidad de tener que apagarlo por las noches 😉.

4.2 Data Lake

Fundamentalmente nuestro Data Lake será un base de Datos Hive montada sobre Hadoop (con HDFS, YATN y MAP REDUCE funcionando).

HIVE nos permitirá tener nuestro "DataWarehouse" distribuido en el clúster de Hadoop (inicialmente un Maestro y tres Slaves levantados y configurados mediante Google DataProc).

HIVE soportara inicialmente una tabla con el histórico de los datos proporcionados por el dataset de Airbnb (una vez filtrado y limpiado) y enriquecido con el crawling de milanuncios.com.

La justificación de usar HIVE y una arquitectura Big Data se hace evidente desde la perspectiva de la ingente cantidad de datos que acumulara nuestra tabla de históricos de apartamentos de Airbnb.

4.3 Service Delivery

Una vez que los informes generados se encuentren en el segmento (bucket de Google Cloud) asociado a nuestro clúster de Hadoop la entrega al cliente se hará por correo mediante una tarea programada en DataProc.

5 Operación

5.1 La operativa del proceso es la siguiente.

1. Descarga diaria del dataset de AIRBNB del siguiente enlace:
   https://public.opendatasoft.com/explore/dataset/airbnb-listings/export/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features&q=Madrid&dataChart=eyJxdWVyaWVzIjpbeyJjaGFydHMiOlt7InR5cGUiOiJjb2x1bW4iLCJmdW5jIjoiQ09VTlQiLCJ5QXhpcyI6Imhvc3RfbGlzdGluZ3NfY291bnQiLCJzY2llbnRpZmljRGlzcGxheSI6dHJ1ZSwiY29sb3IiOiJyYW5nZS1jdXN0b20ifV0sInhBeGlzIjoiY2l0eSIsIm1heHBvaW50cyI6IiIsInRpbWVzY2FsZSI6IiIsInNvcnQiOiIiLCJzZXJpZXNCcmVha2Rvd24iOiJyb29tX3R5cGUiLCJjb25maWciOnsiZGF0YXNldCI6ImFpcmJuYi1saXN0aW5ncyIsIm9wdGlvbnMiOnsiZGlzanVuY3RpdmUuaG9zdF92ZXJpZmljYXRpb25zIjp0cnVlLCJkaXNqdW5jdGl2ZS5hbWVuaXRpZXMiOnRydWUsImRpc2p1bmN0aXZlLmZlYXR1cmVzIjp0cnVlfX19XSwidGltZXNjYWxlIjoiIiwiZGlzcGxheUxlZ2VuZCI6dHJ1ZSwiYWxpZ25Nb250aCI6dHJ1ZX0%3D&location=16,41.38377,2.15774&basemap=jawg.streets
2. Limpieza y filtrado del dataset de Airbnb.
3. Crawling diario de milanuncios.com.
4. Carga diaria de datos en la Base de datos histórica de HIVE.
5. Cada 15 días procesamiento de la información y generación de Informes.
6. Envío por correo de los informes generados a los stakeholders de Airbnb.

6 Desarrollo

6.1 DataSet Airbnb

Sobre el dataset de Airbnb se llevará a cabo un proceso de limpieza y filtrado mediante la herramienta Trifacta a utilizada por nosotros en un modulo anterior. Otras herramientas similares relacionadas con los procesos ETL es Talend, cuyo uso se analizará de cara al futuro y el aumento de datos a limpiar.

6.2 Crawling milanuncios.com

Para llevar a cabo el crawling de milanuncios.com se usa la librería de Python Scrappy. Este proceso es susceptible de ser automatizado mediante un cron en una maquina sin demasiados recursos, podríamos hacerlo incluso con una raspberry.

6.3 Despliegue Cloud

Como ya hemos indicado el despliegue Cloud se lleva a cabo en la plataforma de Google denominada Google Cloud.

Para ello usamos el servicio de la plataforma llamado Dataproc. Este servicio permite desplegar de manera rápida y sencilla un clúster de Hadoop con la configuración que estimemos oportuna (numero de esclavos, potencia de las maquinas del clúster). Además, este servicio ya incluye la instalación de Hive en el despliegue del clúster Hadoop.

Para acceder a la Base de Datos de HIVE podemos hacerlo de dos maneras:

- Desde la Shell que proporciona Google Cloud mediante la herramienta beeline. Ej: beeline -u jdbc:hive2://localhost:10000
- Usando la librería de Python PyHive que proporciona un interfaz a la Base de Datos Hive para llevar a cabo las consultas que necesitemos llevar a cabo.

Desde Google Cloud también podemos configurar los accesos a nuestras maquinas (Firewall) mediante el servicio “Red de VPC” en el que podemos indicar que Ips tienen acceso y a que puertos.

También desde Dataproc se pueden programar y encolar tareas para ser ejecutadas por YARN.

Por ultimo, Dataproc permite asignar un segmento de Storage al clúster que desplegamos. Esto facilita sobre manera el proceso de Stagin, que veremos mas abajo, así como el almacenamiento de los informes generados.

6.4 Staging

Como se ha indicado en el apartado anterior, Dataproc permite asociar un segmento de Storage al clúster Hadoop desplegado. Este segmento es una especie de bucket (similar al S3 de Amazon

Web Service), que esta en la misma subred del clúster y que por tanto permite conectarnos con el desde nuestro maestro de Hadoop para el intercambio de ficheros.

Además, permite guardar allí la información que no queremos perder cuando eliminamos o detenemos el clúster de Hadoop

De esta manera es Stagin se convierte en un proceso tan sencillo como pulsar sobre el botón de "Subir fichero", de igual modo la salida de cualquier tarea llevada a cabo se puede redirigir a dicho segmento.

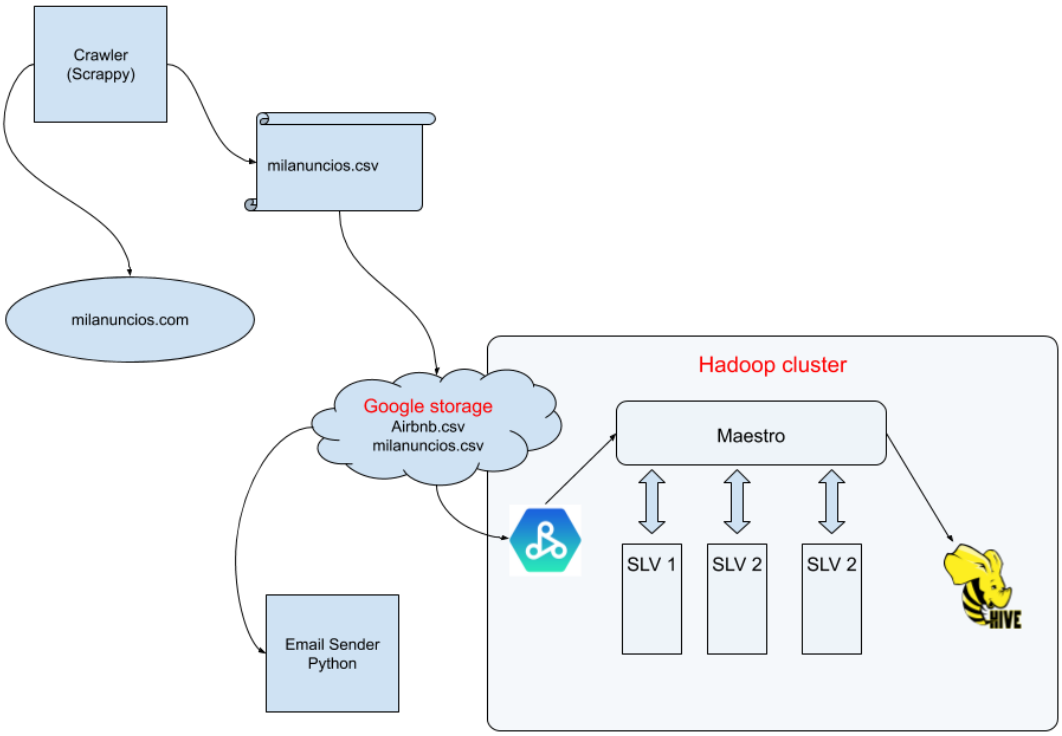
Para la correcta gestión del Staging se creará, en el segmento de Google Cloud un directorio de entrada llamado **input**, donde se meterán el dataset de Airbnb y el crawling de milanuncios.com y otro de salida llamado **output** donde se guardarán los informes generados, para poder ser enviados por correo.

6.5 Generación de Informes

La generación de los informes en base a la información que se va acumulando en la Base de Datos de HIVE se puede llevar a cabo de múltiples maneras:

- Mediante Jobs de YARN currándonos nuestros jar correspondientes una vez que tengamos la destreza necesaria.
- Mediante script de Python usando la librería PyHive que esta diseñada como un conector a HIVE.

7 Diagrama


Crawler
(Scrappy)
milanuncios.csv
milanuncios.com
Google storage
Airbnb.csv
milanuncios.csv
Hadoop cluster
Maestro
SLV 1
SLV 2
SLV 2
HIVE
Email Sender
Python