Motor de Búsqueda usando Hadoop y algoritmos: Índice Invertido y Pagerank

Patrick Xavier Marquez Choque*
Universidad Católica San Pablo

Jean Carlo Cornejo Cornejo†
Universidad Católica San Pablo

Mayo 2022

1 Desarrollo

Para la implementación de dicho trabajo, se nos dió la tarea de implementar un clúster en *Hadoop* y implementar el algoritmo de índice invertido y *page rank* para lograr un motor de búsqueda de términos de data dentro de un sistema distribuido, así que como decisión de grupo se decidió utilizar el sistema de Cloud de Google : *Google Cloud Platform*.

El repositorio del proyecto del Motor de Búsqueda se encuentro en GitHub en el enlace [1]

Para realizar este proyecto se tomaron los siguientes puntos:

*e-mail:patrick.marquez@ucsp.edu.pe
†e-mail:jean.cornejo@ucsp.edu.pe
[1] `https://github.com/patrick03524/Search-Engine-Inverted-Index-Page-Rank-Cloud-Computing`.

1.1 Configuración del Cluster dentro de *Google Cloud Platform*

Para este proyecto se logró una correcta configuración de un cluster utilizando *Google Cloud Platform*, se utilizó principalmente este tutorial: [2]

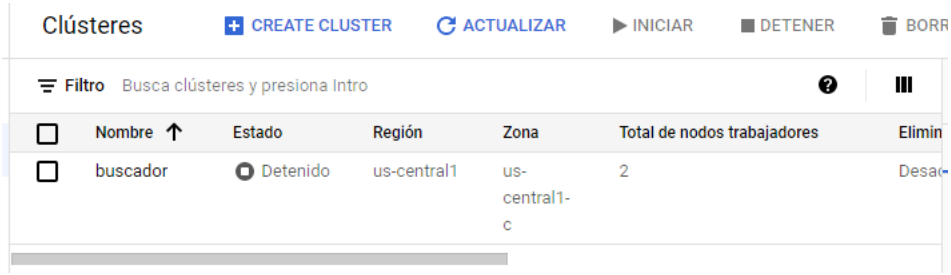

Figure 1: Vista principal del cluster configurado.

Lo que se logró con esto es la creación de un cluster distribuido en *Google Cloud Platform* como se puede apreciar en la Figura 1 de tal manera que dentro de este cluster exista un nodo maestro y 2 nodos esclavos de la siguiente manera:

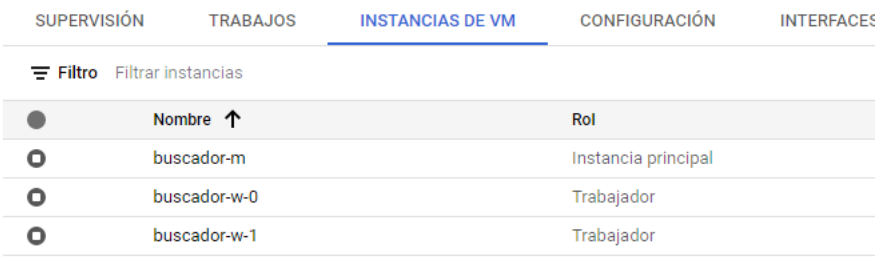

Figure 2: Vista de los Nodos del cluster.

Posteriormente se realizó una configuración de las instancias de las máquinas virtuales de todos los nodos para la configuración del protocolo de comunicación SSH, configuración de las variables de entorno para Java e instalación de todas las librerías de Apache Hadoop para la compilación del proyecto del Motor de Búsqueda.

[2] `http://www-scf.usc.edu/~shin630/Youngmin/files/HadoopInvertedIndexV5.pdf`.

Por último luego de la configuración y actualización de los sub-sistemas de Linux de nuestras instancias de las Máquinas virtuales, se realizó un lanzamiento de un código de prueba para comprobar que todo funcionará; así que se lanzó una tarea de compilación de un archivo .java dentro del nodo maestro, esto resultó en un archivo .jar así que esto fue enviado con el comando "hadoop fs" para enviarlo a un bucket que nos proporciona *Google Cloud Platform* con el nombre de "dataproc" para cargar nuestra data de manera local. Esto se realizó de igual manera para la compilación del proyecto de Motor de Búsqueda.

1.2 Implementación del Motor de Búsqueda

Dentro de la implementación del Motor de Búsqueda se crearon archivos en Java de clases con el algoritmo de Indice Invertido y el Buscador como tal. Cada clase tenía sus funciones Map y Reduce para lograr la utilización de MapReduce dentro del entorno distribuido en nuestro Clúster. Debido a ciertos problemas no se logró una correcta implementación de los códigos en Java pero los intentos de este proyecto se encuentran en el repositorio adjunto de Github.

Adicionalmente, en el trabajo era necesario añadir como complemento una sección visual de los resultados de nuestros algoritmos, a manera de mostrar un resultado, se ha de resaltar que debido a ciertos problemas con el código y la falta de fiabilidad en los resultados por la incorrecta implementación de las secciones correspondientes al output, no se progresó en la idea original, la cual era usar un framework de desarrollo web y un medio de comunicación para obtener los resultados.

En la imagen 3 se observa los códigos presentes en el sistema de archivos de Google y preparados para ser enviados a Hadoop en la forma de jobs.

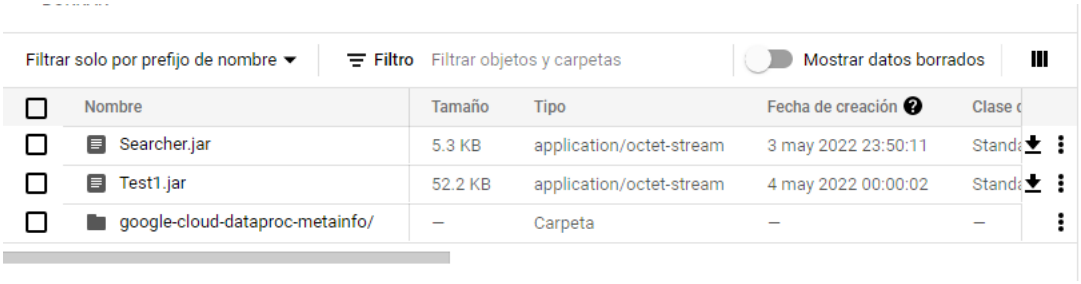

Figure 3: Códigos incluidos en el clúster preparados para ejecutar jobs.

En la imagen 4 se encuentra el resultado luego de ejecutar alguno de los .jar previamente mostrados dados un tiempo de ejecución.

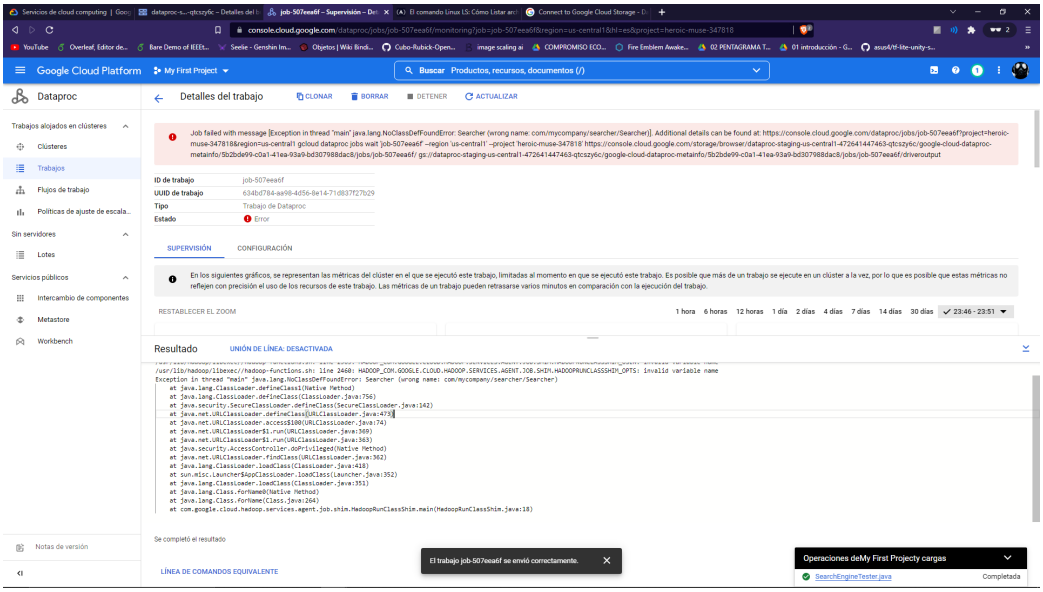

Figure 4: Error mostrado a la hora de ejecutar jobs.

1.3 Conclusiones

Como principal conclusión de este trabajo podemos afirmar que la utilización de un sistema distribuido para realizar algoritmos con una gran cantidad de datos es la manera más óptima de realizar y gracias a plataformas como la que provee Google, se nos permite una rápida configuración del cluster como tal para realizar estas tareas.

Posteriormente podemos destacar las dificultades de la implementación de los algoritmos ya que este paradigma distribuido es bastante complicado de manejar pero que provee bastantes ventajas para procesar mucha información. De igual manera *Google Cloud Platform* provee un sistema confiable y robusto con el que se pueden crear algoritmos del Indice Invertido y Page Rank de manera mucho más eficiente.