Comparación de la capacidad de respuesta entre PostgreSQL y el Stack ELK en un entorno Docker

Lic. Jonathan Fernando Romano - Ing. Agr. Juan Manuel Alonso
Junio 2021

***Resumen*—Se analizó una base de datos de 355.000 registros de Documentos de tránsito vegetal (DTV-E) emitidos por el Servicio Nacional de Sanidad y Calidad Agroalimentaria (SENASA) de Argentina para el traslado de plantas de vivero. Se montó un entorno Docker en Linux a los efectos de comparar la capacidad de respuesta de los motores de base de datos PostGreSQL y ElastiSearch en función de consultas de menor a mayor complejidad. FALTAN CONCLUSIONES**

***Index Terms*— Docker, PostGreSQL, Elasticsearch, Logstash, Kiban., Comparativa.**

I. INTRODUCCION

En el marco del curso de posgrado "Captura y Almacenamiento de información" dictado por la Facultad de Informática de la Universidad de La Plata, se realizó el presente trabajo de investigación a los efectos de aplicar los conocimientos adquiridos durante la cursada. En este sentido, se tomó como punto de partida una base de datos del Programa Nacional de Sanidad de Material de Propagación del Servicio Nacional de Sanidad y Calidad Agroalimentaria (SENASA) perteneciente a movimientos de plantas de vivero del año 2016 a 2021 por todo el territorio nacional argentino.

El sistema Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software. Permite un montaje ágil de los diferentes entornos y configuraciones, lo que hace esta herramienta ideal para establecer las comparaciones deseadas. Se montó un ambiente de prueba sobre un Sistema Operativo Ubuntu 20.04, utilizando el entorno Docker para las instancias de las bases de datos a testear.

Durante el desarrollo de este trabajo, se describirá la base de datos a analizar, como fue el montaje del entorno Docker y sus complementos, intentando detallar los aciertos y desaciertos que tuvimos durante todo el proceso.

II. OBJETIVOS

- Aplicar los conocimientos vistos en la materia de Almacenamiento y Captura de la Información, como por ejemplo, Imágenes y Contenedores en Docker, Arquitectura de Software, Docker-Compose y diferentes Herramientas para la extracción y explotación de los datos.

- Montar un ambiente en Docker que contenga, un DBMS (Sistema de gestión de Base de datos) PostgreSQL y el Stack ELK (Elasticsearch, Logstash, Kibana).

- Evaluar el rendimiento en tiempos de respuesta sobre el procesamiento de consultas de diferente complejidad entre ambas herramientas.

III. MATERIALES Y METODOS

A. Base de datos (BD)

La base de datos a analizar corresponde a 355.000 registros distribuidos en 21 campos de datos de Documentos de tránsito Vegetal Electrónicos (DTV-e) emitidos por viveristas de la Argentina en cumplimiento a normativas nacionales y con el fin de amparar el tránsito de plantas entre distintos puntos del país.

El archivo original importado esta en formato .CSV (del inglés comma-separated values) y tiene un peso de 109 MB (114.644.979 bytes).

B. Configuración y preparación del entorno.

En primer lugar se pensó en montar una máquina virtual con sistema operativo Ubuntu Linux, pero por limitaciones de hardware se tuvo que migrar el montaje a una partición de disco duro e instalación convencional. A partir de allí se pudieron correr los procesos sin inconvenientes y a una velocidad considerablemente mayor.

Durante el proceso aparecieron algunas incompatibilidades que tuvimos que superar, por ejemplo la versión de Java que teníamos instalada en la distribución de Ubuntu era diferente a la que necesitaba el Logstash. Aunque teníamos una versión más moderna hubo que hacer un downgrade a la versión 8. hzzo perderra que todo funcione. Este tipo de inconvenientes

Otro punto importante, fue la migración de la base de datos PostgreSQL al entorno Kibana con Elastic Search y el armado de los archivos de configuración correspondientes. Una vez montada la base de datos en PostgreSQL y en ELK, el manejo de los entornos web en fue sencillo.

El paso a paso detallado del montaje del entorno Docker, como de los paquetes de Docker compose, PostGreSQL, PgaAdmin4, Elastic Search, Logstash y Kibana, se pueden consultar en el readme.md disponible en: https://github.com/warasoft/tp_final#readme

C. Matriz de comparación de tiempos de respuesta.

Se elaboró una matriz de 3 x 2 para evaluar la respuesta de los distintos motores en función de 3 consultas de menor a mayor complejidad

La consulta de baja complejidad consiste en traer toda la base de datos sin hacer ningún filtrado. La consulta de mediana complejidad, es un filtrado por tipo de movimiento en donde se seleccionan todos los documentos emitidos bajo el tipo "Vivero - Vivero". Por último, la de alta complejidad tiene dos condiciones a cumplir, debe mostrar todos los documentos emitidos bajo el tipo "Vivero - Vivero" con origen la provincia de Buenos Aires (Filtro por “Tipo de movimiento” y campo “O_provincia”).

La elección de diferentes tipos de consultas de BD busca abarcar la mayor cantidad de situaciones posibles, de forma tal de poder determinar el comportamiento de cada una de las herramientas de forma abarcativa y no con un sesgo que pueda favorecer una u otra.

IV. RESULTADOS

A continuación en la Tabla 1 se exponen los resultados de los tiempos de respuesta en primera ejecución para los motores de búsqueda analizados según las condiciones establecidas en la matriz.

En la Tabla II se muestra la diferencia que existe en tener los datos cargados en memoria en una segunda ejecución.

TABLA I

RESULTADOS DE PERFORMANCE DE LAS HERRAMIENTAS ELEGIDAS PARA LAS DIFERENTES CONDICIONES

	Tiempos	
Tipo Consulta	PGS	ELK
Simple	6 secs 682 msec	912 msec
Media	1 sec 262 msec	1549 msec
Alta	1 sec 9 msec	931 msec

*PGS= PostgreSQL, EKL= Elastic Search, Kibana, Logstash

TABLA II

RESULTADOS DE PERFORMANCE EN UNA SEGUNDA EJECUCIÓN DE QUERYS (POSTGRES) Y FILTROS (ELK)

	Tiempos	
Tipo Consulta	PGS	ELK
Simple	2 secs 464 msec	358 msec
Media	1 sec 253 msec	698 msec
Alta	1 sec 3 msec	468 msec

V. CONCLUSIONES

Como primera conclusión se puede decir que las últimas versiones del Stack ELK, requieren mayor capacidad de Hardware. Se pudo probar la versión 7.13.1 EKL, con una pc dedicada con 4Gb de RAM, Microprocesador Core I5, y sin embargo al levantar todo el ambiente de trabajo, en el momento de transferir los datos del Postgresql al elasticsearch, el sistema operativo, NO respondía, siendo solo 355.000 registros aproximadamente.

Con la versión 6.6.0 del Stack ELK, se pudo trabajar sin inconvenientes. Por ejemplo, en la trasferencia de 355.000 registros de Postgresql al elasticsearch con logstash, demoró solo 3 min, 42 sec, 55 msec.

Desde el punto de vista de la performance se comprobó que la herramienta Stack ELK fue más eficiente en términos de tiempo de respuesta que el gestor de BD PostgreSQL como muestran la Tabla I y Tabla II. A su vez, con los datos cargados en memoria, existe una notable mejora en ambos sistemas.

Por último y no menos importante, es la interacción con el usuario final de los datos, siendo KIBANA del ELK una forma más amigable a la hora de hacer las consultas pre-establecidas que diariamente se pueden utilizar. Esto significa una curva de aprendizaje menos compleja que trabajar directamente con consultas del tipo Sql. Todo dependiendo del puesto de trabajo del usuario.

REFERENCES

A. Cotten. (2021, May.). Elastic stack (ELK) on Docker. GitHub. [Online].

ENLACES DE INTÉRES

https://github.com/caas/docker-elk.git
https://docker.com/
https://hub.docker.com/
https://www.elastic.co/es/
https://www.elastic.co/es/downloads/past-releases/logstash-6-6-0

AUTORES

ALONSO, Juan Manuel. Ingeniero agrónomo, recibido en la Universidad de Morón en el año 2012 y con un posgrado en Alta Dirección de Agronegocios y Alimentos de la Facultad de Ciencias Agrarias de la Universidad de Buenos Aires en el año 2018. Desde el año 2020, se encuentra estudiando la Maestría de inteligencia de datos orientada en Big Data de la Universidad Nacional de La Plata.

Actualmente trabaja como Analista Profesional en la Dirección Nacional de Protección Vegetal del SENASA en la coordinación del Programa Nacional de Sanidad de Material de propagación Vegetal.

ROMANO, Jonathan Fernando. Licenciado en Informática, recibido en la Universidad de Palermo, año 2013. Actualmente se encuentra cursando la Carrera de Esp. en Inteligencia de Datos Orientada a Big Data.

Trabaja en la ARMADA ARGENTINA, como Ayudante del Jefe de División Sistemas del Servicio Administrativo Financiero de la Armada.