Joxan Andrey Fuertes Villegas-2021128656

Data in: documents and Indices

Elatic Search es un almacen distribuido de documentos. No guarda los documentos como una fila de columnas de datos, sino que utiliza estructuras de datos complejas serializadas en formato JSON. Elastic Search utiliza una estructura llamada ***índice invertido*** que permite búsquedas rápidas de textos completos, esta estructura identifica una palabra única de cualquier texto e identifica todos los documentos en los que dicha palabra aparece. Un índice es una colección de documentos optimizada, un documento es una colección de campos y cada campo es una pareja llave-valor que contiene los datos.

Elastic Search indexa los datos de cada campo según su tipo, texto-***índice invertido***, números-***K-dimensional Binary Tree***. Justo esta habilidad de ordenar los datos por campo es lo que le otorga su velocidad. Elastic Search no necesita por que detectará y mapeará automáticamente enteros, booleanos, flotantes, etc en los tipos apropiados de Elastic Search. Sin embargo también existe la opción de definir manualmente cómo se va a manejar el mapeo (cómo los campos va a ser almacenados e indexados).

Information out: search and analyze

Si bien se puede usar a elastic Search para almacenar y devolver documentos, la verdadera cualidad se encuentra en la suite completa de capacidades de búsqueda del motor de búsqueda de la librería apache Lucene. Elastic Search provee una API sin estado para manejar nuestro cluster, indexar y buscar datos. Esta API soporta:

- consultas de texto completo: devuelven los documentos donde aparece el string buscado según prioridad.
- consultas estructuradas: donde se pueden buscar atributos específicos (género y edad) de cierto índice (empleado) ordenados por un atributo (fecha de nacimiento descendente), similares a SQL.
- consultas complejas: una combinación de las anteriores.

Estas capacidades se pueden accesar utilizando el lenguaje de consultas basado en JSON de Elastic Search. Este estilo de consultas se utilizan para buscar o agregar datos nativamente dentro de elastic search. Utilizando ODBC y JDBC se puede usar SQL para interactuar directamente con Elastic Search.

Elastic Search además de encontrar el elemento de búsqueda, permite realizar un análisis de patrones, tendencias y metricas clave. Podría responder preguntas como ¿Cuál es el largo promedio de la aguja que buscás en el pajar? o preguntas más sutiles como ¿Cuál es el fabricante de agujas más popular? Además, este análisis se produce en tiempo en tiempo real, junto a la búsqueda, esto lo hace igualmente veloz ya que se maneja con el mismo sistema de estructuras de las búsquedas. Además de esto se puede incluir ***Machine Learning*** para automatizar el proceso de creación de estadísticas y tener una referencia confiable del comportamiento de los datos. Esto permite detectar anomalías de datos y anormalidades en algún miembro de la población.

Scalability and resilience.

Elastic Search es capaz de crecer según las necesidades del programador. Se pueden agregar servidores (nodos) a un clúster y elastic search distribuirá ***fragmentos (shards)*** (índice contenido en sí mismo) a múltiples nodos. Lo anterior crea redundancia lo que proteje al sistema de fallas en el hardware y aumenta la capacidad de respuesta a consultas. Según el clúster crezca o decrezca, Elastic Search reequlibrará y migrará los fragmentos. Hay 2 tipos de fragmentos, los principales y las réplicas. Los principales son fijos y no desaparecen según el tamaño del clúster, las réplicas son copias de los principales y son los que permiten el reajuste.

Hay ciertas consideraciones a tomar en cuenta con el tamaño y la cantidad de fragmentos principales. Mientras más fragmentos principales haya más sobrecarga se generará para mantenerlos, y mientras más grande el tamaño de los fragmentos, más se dura en trasladorlos a través del clúster. Hacer fragmentos más pequeños para dejarle el problema a las consultas tampoco es una buena solución, por que se está redirigiendo la misma carga, ahora a las consultas.

Para mantener alta disponibilidad entre los nodos de un clúster, hay que considerar el caso en el que un servidor falle. ¿Cómo se resolvería? Con ***Cross-Cluster Replication (CCR)*** o replicación de clúster cruzada. Lo anterior permite sincronizar los índices de tu clúster primario, a un clúster remoto. Esta práctica también sirve para leer peticiones de usuario con una mayor proximidad, no solo en caso de falla. Sin embargo el clúster primario es el único que puede manejar peticiones de escritura, el secundario solo maneja de lectura.

Para poder monitoriar, asegurar y manejar el clúster de Elastic Search, este nos permite usar kibana como un centro de control. funciones como ***data rollups*** (paquetes acumulativos de datos) y ***index life cycle management*** (Gestión del ciclo de vida de los índices) nos permiten manejar inteligentemente nuestros datos a través del tiempo.

fuente: Elastic Search Guide