TRABAJO PRÁCTICO FINAL

Alternativas a la Persistencia Relacional

Universidad Nacional de Río Negro - Licenciatura en Sistemas - Base de Datos II

Alumno: Braian Lisandro Graff - Legajo 1167

Docente: Enrique Molinari

Fecha: 20/02/2024

Mesa: Segundo llamado del turno Febrero 2024

Instancia: Regular

RÍO NEGRO
UNIVERSIDAD
NACIONAL

Índice

Introducción

En este TP Usted debe realizar la implementación del Blog (TP5) utilizando una base de datos NoSQL (no puede ser MongoDB).

Pueden utilizar una base de datos de clave-valor, documentos o basada en columnas. Pueden buscar e investigar la base de datos a utilizar ingresando a http://nosql-database.org/.

Puede suceder que la base de datos que elija no soporte todas las funciones que necesita el blog para funcionar. Por ejemplo, la búsqueda de texto libre o la sumarización de cantidad de posts por autor. Si esto es así, pueden utilizar la base de datos igual, pero deben agregarle una instalación en cluster (dos nodos mínimo). Y mostrar que bajando un nodo el blog sigue funcionando. Todas las APIs de recuperación de post o páginas por ID deben estar implementadas.

Por ejemplo, Redis es una base de datos de tipo clave/valor, donde no van a poder implementar ciertas características. ElasticSearch es una base de datos basada en documentos, con excelentes funciones de búsqueda. Con esta base de datos van a poder implementar todas las características del blog.

Además de mostrar la aplicación funcionando, y explicar cómo se implementa cada API, se pide:

- Documentación básica de cómo se instala e inicia y detiene el servicio.

- Explicar brevemente qué tipo de base de datos es, que tipo de replicación tiene, como se define según el teorema de CAP y PACELC. Indique las fuentes de dónde obtuvo esta información.

Base de datos no relacional

Las bases de datos no relacionales, también conocidas como NoSQL, almacenan y consultan los datos en estructuras diferentes a las bases de datos relacionales. En lugar de utilizar la típica estructura de una base de datos relacional, las bases de datos NoSQL alojan los datos dentro de una estructura de datos como documentos JSON, como un grafo, o simplemente como pares clave/valor, de manera no relacional.

Este diseño de base de datos no relacional no requiere un esquema y ofrece una rápida escalabilidad para gestionar grandes conjuntos de datos normalmente no estructurados.

NoSQL es también un tipo de base de datos distribuida, lo que significa que la información se copia y almacena en varios servidores, que pueden ser remotos o locales. De esta manera, se garantiza la disponibilidad y la fiabilidad de los datos. Si alguno de los datos queda fuera de línea, el resto de la base de datos puede seguir ejecutándose.

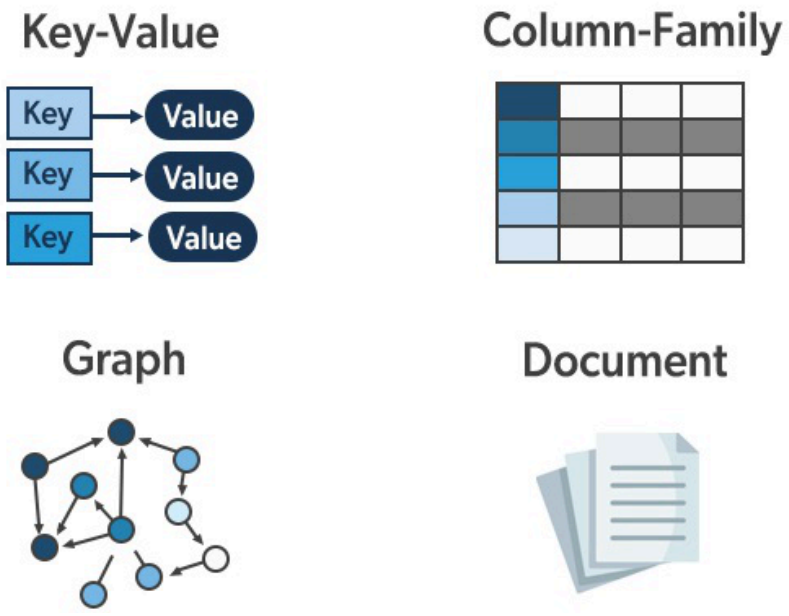

Imagen 1 - Tipos de bases de datos NoSQL

Tipos de bases de datos NoSQL

Almacén de pares clave-valor

Este modelo de datos sin esquema se organiza en un diccionario de pares clave-valor, donde cada elemento tiene una clave y un valor. La clave puede ser algo similar a los que encontramos en una base de datos SQL, como un ID de carro de compras, mientras que el valor es una matriz de datos, como cada artículo individual en el carro de compra de ese usuario.

Se utiliza comúnmente para almacenar en caché y guardar información de sesión del usuario. Sin embargo, no es ideal cuando se necesita extraer varios registros a la vez.

Almacén de documentos

Como su nombre lo indica, las bases de datos de documentos almacenan datos como documentos. Pueden ser útiles en la gestión de datos semiestructurados y, por lo general, los datos se almacenan en formato JSON, XML o BSON. Esto mantiene los datos juntos cuando se utilizan en aplicaciones, de modo que se reduce la cantidad de conversión necesario para utilizarlos.

Se obtiene más flexibilidad ya que los esquemas de datos no tienen que coincidir entre documentos (ejemplo, nombre vs primer_nombre). Sin embargo, esto puede ser problemático para transacciones complejas, ya que puede generar corrupción de datos.

Almacén distribuido en columnas

Estas bases de datos almacenan información en columnas y filas, se puede parecer mucho a una base de datos relacional, al menos desde el punto de vista conceptual. La potencia real de una base de datos de la familia de columnas radica en su enfoque desnormalizado para la estructuración de datos dispersos, que viene del enfoque orientado hacia las columnas para el almacenamiento de datos.

Este tipo de base se ha diseñado para gestionar grandes cantidades de datos en varios servidores y agrupaciones en clúster que abarcan varios centros de datos. Se utilizan para diversos casos de uso, como sitios web de redes sociales y análisis de datos en tiempo real.

Almacén en grafos

Este tipo de base de datos normalmente alberga datos de un grafo de conocimiento. Los elementos de datos se almacenan como nodos, aristas y propiedades. Cualquier objeto, ubicación o persona puede ser un nodo. Una arista define la relación entre nodos.

El propósito de un almacén de datos de grafos es permitir a una aplicación realizar consultas de manera eficaz que recorran la red de nodos, aristas y analizar las relaciones entre entidades.

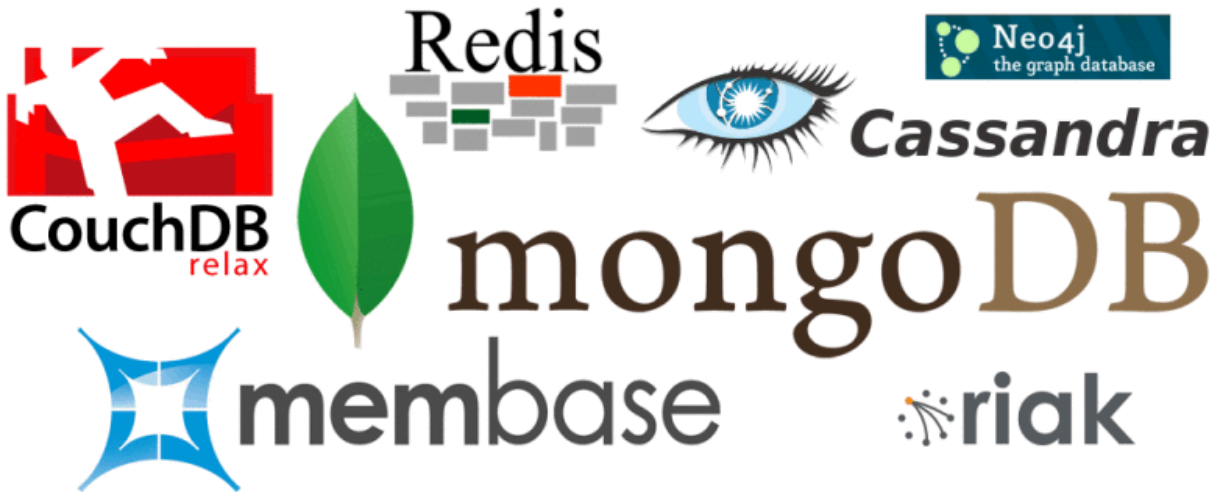

Imagen 2 - Algunas BD NoSQL

Implementación

Para resolver el trabajo práctico elegí Elasticsearch, una base de datos NoSQL orientada a documentos, ya que soporta todas las funciones que necesita el blog para funcionar.

Elasticsearch es un motor de búsqueda y analitica que permite realizar búsquedas de manera muy rápida, de ahí que comúnmente se utilice como herramienta de búsqueda en aplicaciones y sitios web, pero también como herramienta de login, monitoreo, analitica de negocio y seguridad entre otras cosas. Es open source y distribuida, lo que significa que podemos escalar elasticsearch de forma horizontal.

Instalación de Elasticsearch

Luego de verificar la compatibilidad de mi sistema operativo con Elasticsearch opte por instalarlo y configurarlo con docker en una notebook con sistema operativo Linux, más precisamente Ubuntu 20.04. En ubuntu se puede instalar la versión de Elasticsearch 8.3.X en adelante, actualmente la última versión estable de Elasticsearch es la 8.12.1.

Para poder utilizar la imagen docker de Elasticsearch sin problemas tuve que modificar una propiedad del kernel para aumentar la memoria virtual (vm.max_map_count). Comando para linux: `sysctl -w vm.max_map_count=262144`

Las pasos para iniciar un clúster de Elasticsearch de un solo nodo con docker en linux son:

1. Instalar docker:
    1.1. Descargar el paquete .deb más reciente de docker.
    1.2. Instalar el paquete con apt:

```
sudo apt-get update

sudo apt-get install ./docker-desktop-<version>.deb
```

2. Crear una nueva red en docker llamada elastic:

```
docker network create elastic
```

3. Descargar la imagen de Elasticsearch Docker:

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.12.1
```

4. Iniciar el contenedor Elasticsearch:

```
sudo docker run --name es01 --net elastic -p 9200:9200 -it -m 1GB
docker.elastic.co/elasticsearch/elasticsearch:8.12.1
```

5. Buscar password y certificados que genera Elasticsearch en el log, si queremos podemos generarlos nuevamente el comandos:

```
docker exec -it es01
/usr/share/elasticsearch/bin/elasticsearch-reset-password -u
elastic
```

6. Como no estamos en producción y solo es con fines educativos, para no tener que configurar los certificados de seguridad, elimine la conexión segura:
   6.1. Extraemos el archivo elasticsearch.yml del contenedor a nuestra pc:

```
sudo docker cp
es01:/usr/share/elasticsearch/config/elasticsearch.yml .
```

   6.2. Editamos el archivo elasticsearch.yml extraído, todas las líneas que están en true la pasamos a false.
   6.3. Importamo el archivo elasticsearch.yml editado al contenedor:

```
sudo docker cp elasticsearch.yml
es01:/usr/share/elasticsearch/config/elasticsearch.yml
```

Instalación del Cliente Java

El Blog consume los datos de un backend hecho en java con Spark para publicar servicios web (API Web). Este backend es proporcionado en el trabajo práctico, por tal motivo para consumir la API Rest de Elasticsearch necesitamos utilizar el cliente oficial de java para Elasticsearch.

Como requisitos para instalar el cliente necesitas tener java 8 o posterior y una biblioteca de mapeo de objetos JSON para permitir una integración perfecta con la API de Elasticsearch.

Para instalar el cliente en el proyecto necesitamos editar el archivo pom.xml agregando las siguientes líneas:

```
<dependency>
      <groupId>co.elastic.clients</groupId>
      <artifactId>elasticsearch-java</artifactId>
      <version>8.12.0</version>
</dependency>
<dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.12.3</version>
</dependency>
```

Instalación Jakarta

Luego de configurar las dependencias, si la aplicación falla con el error `ClassNotFoundException: jakarta.json.spi.JsonProvider` es necesario agregar explícitamente de dependencia Jakarta al archivo pom.xml

```
<dependency>
    <groupId>jakarta.json</groupId>
    <artifactId>jakarta.json-api</artifactId>
    <version>2.0.1</version>
</dependency>
```

Conexión a la Base de Datos

El cliente Java está estructurado en torno a tres componentes principales:

- **Clases de cliente API:** Métodos y estructura de datos fuertemente tipados para las API de Elasticsearch.
- **Un mapeador de objetos JSON**: Para asignar las clases de la aplicación a JSON y integrarlas perfectamente con el cliente.
- **Una implementación de capa de transporte**: Para el manejo de solicitudes HTTP.

```
RestClient restClient = RestClient.builder(HttpHost.create("http://localhost:9200")).build();
ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());
ElasticsearchClient esClient = new ElasticsearchClient(transport);
```

Imagen 3 - Conexión Cliente Elasticsearch

Implementación de la API Web

El desarrollo de la API Web lo modularize en tres partes:

- Creación del índice **pagina** y la inserción de dos documentos páginas para el blog.
- Creación de índice **post** y la inserción de siete documentos post.
- Conexión e implementación de los métodos para que consuma el blog.
    - /pagina-id/:id
    - /ultimos4posts
    - /post-id/:id
    - /byautor
    - /posts-autor/:nombreautor
    - /search/:text

En esta sección no voy a entrar en detalle porque se puede ver la implementación en el proyecto adjunto.

Imagen 4 - Blog

Breve explicación Elasticsearch

Elasticsearch se desarrolló para ser distribuida y fácil de escalar horizontalmente para ocuparse de enormes cantidades de datos.

En cuanto a la tolerancia a particiones, disponibilidad y consistencia, Elasticsearch es un sistema CP (consistencia), pero te permite lograr un comportamiento AP (disponibilidad) si es necesario.

En términos de escalado, un índice se divide en uno o más shards. Esto se especifica cuando se crea el índice y no puede modificarse. Por lo tanto, un índice debe dividirse en shards de forma proporcionada con el crecimiento anticipado. A medida que se agrega más nodos a un cluster de Elasticsearch, hace un buen trabajo reasignando y acomodando los shards. Esto lo hace muy fácil de escalar horizontalmente.

Elasticsearch no tiene transacciones en el sentido tipico, no hay forma de revertir un documento enviado, tampo puedes enviear un grupo de documentos y hacer que todos o ninguno se indexen (atomico). Sin embargo, si posee un log de escritura anticipada para garantizar la durabilidad de las operaciones sin necesidad de una costosa confirmacion. Tambien se puede especificar el nivel de consistencia de las operaciones de indexacion, en cuanto a la cantidad de replicas que deben reconocer la operacion antes de la devolucion. De froma predeterminada, es la mayoria, es decir; (N/2) + 1.

El control de concurrenciaoptimista se realiza especificando la version de los documentos enviados.

Bibliografía

- IBM. ¿Qué es una base de datos NoSQL? https://www.ibm.com/es-es/topics/nosql-databases
- ORACLE. ¿Qué es NoSQL? https://www.oracle.com/ar/database/nosql/what-is-nosql/
- Elastic. Elasticsearch Guide [8.12] https://www.elastic.co/guide/index.html
- Brasetvik, Alex. Elasticsearch como base de datos NoSQL. https://www.elastic.co/es/blog/found-elasticsearch-as-nosql
- Capdevila, Gabriel. Replicación de bases de datos NoSQL en dispositivos móviles https://sedici.unlp.edu.ar/bitstream/handle/10915/48085/Documento_completo__.pdf?sequence=1&isAllowed=y

Anexos

Se adjuntan en el mismo directorio, junto a este documento pdf los siguientes documentos:

- Documento pdf del trabajo práctico N 5 - MongoDB - Construyendo un Blog
- Carpeta que contiene el frontend del Blog
- Carpeta que contiene el backend del Blog