

Práctica 4

Base de Datos Cassandra

Facultad De Ingeniería, Universidad De Cuenca
BIG DATA

Freddy L. Abad L.

freddy.abadl@ucuenca.edu.ec

Resumen- El incremento exponencial de generación de datos a presentado múltiples problemas a nivel de bases de datos relacionales. Estos problemas se ven desbordados en cuanto a tiempos de respuesta, organización de datos, entre otros. Una de las posibles soluciones existentes son las bases de datos NoSQL, esta tiene diversos enfoques, dependiendo el dominio del problema a resolver. Las Bases de Datos NoSQL clave valor tienen diversas ventajas y son usadas en múltiples campos, este artículo detalla sobre estas bases de datos, enfocando su desarrollo con Cassandra.

Palabras Clave- NoSQL, Neo4j, Grafos, Angular.

I. INTRODUCCIÓN

Las bases de datos NoSQL son ampliamente usadas en la actualidad debido a sus ventajas que brindan a problemas que surgen en bases de datos relacionales. La flexibilidad otorgada por su falta de esquema, permite desarrollar conceptos complejos de las aplicaciones actuales[1]. Las bases de datos NoSQL, se pueden dividir por la tipología de construcción: bases de datos documentales, clave-valor, dirigido por ancha y dirigidos por grafos [2]. El presente informe trata acerca de las bases de datos clave-valor, centrando su desarrollo en la base de datos Cassandra, la cual es una base de datos distribuida de software libre que pertenece a la Apache Software Fundación, su construcción se basa Java [3], por tal es multiplataforma debido a su JDK.

Objetivos

a. Objetivo General

- Implementar la capa de almacenamiento de una aplicación de una base de datos NoSQL clave-valor

b. Objetivo Especifico

- Analizar las funcionalidades que otorga una base de datos clave-valor
- Implementar una aplicación por comandos usando la suite de Cassandra

II. MARCO TEORICO

A continuación, se desarrolla los tópicos tratados en la metodología (Sección III) de la práctica de bases de datos clave-valor.

Base de Datos: Estas se definen como un conjunto de información organizada con determinada relación entre sí, de tal manera que se establezca de manera grupal o estructurada[4]. El objetivo de estas es ser recolectados y explotados por los sistemas informáticos a nivel empresarial [2]. Las bases de datos se clasifican en relacionales o no

relacionales, su implementación depende grandemente a la cantidad de datos que se manejarán, los entornos donde se necesita desarrollasen y la forma como se busca estructurar la data. Ver imagen 1

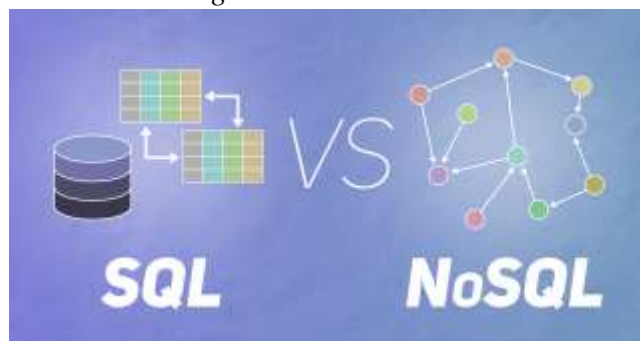


Imagen 1: Esquema de organización de las bases de datos relacional vs no relacional

Base de Datos NoSQL: Estas bases de datos no se establecen bajo relaciones fijas, es decir no necesita manejar tablas obligatoriamente. Son conocidas también como bases de datos no relacionales. Usualmente se emplean en casos donde la flexibilidad, la necesidad de rápido desempeñano, y de consultas recurrentes de escritura y lectura [3][2]. La característica diferenciadora son las propiedades CAP, a diferencia de las ACID [5] de la bases relacionales (Ver imagen 2 y 3).



Imagen 2: ACID

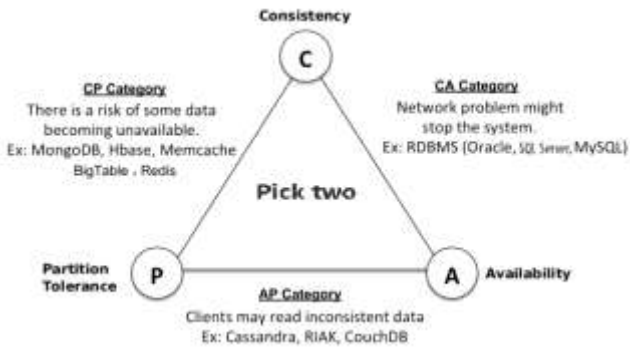


Imagen 3: Teorema CAP

Base de Datos Clave Valor: Estas son bases de datos NoSQL “que utilizan un método simple de clave-valor para almacenar datos. Estas almacenan datos como un conjunto de pares clave-valor en los que una clave sirve como un identificador único. Tanto las claves como los valores pueden ser desde objetos simples hasta objetos compuestos complejos. Las bases de datos clave-valor son altamente divisibles y permiten el escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar” [6][7].

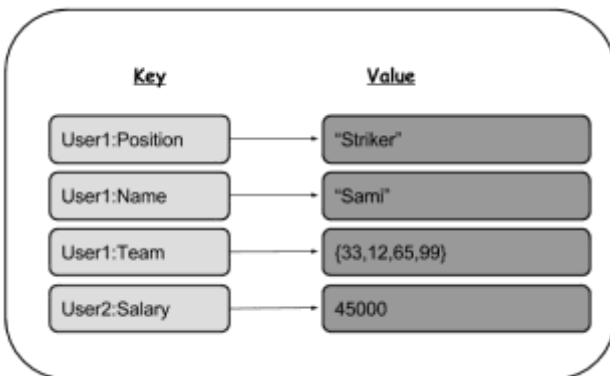


Imagen 4: Esquema de organización clave-valor

Cassandra: Es una base de datos no relacional orientada a clave y valor que proporciona tolerancia a fallos y disponibilidad (AP en el teorema CAP), su desventaja radica en que la consistencia deja de ser una prioridad. Esta orientación en el teorema CAP se debe a que la inserción de datos no es una característica sobresaliente, a diferencia de su eficiencia en consulta de información de forma rápida. La escalabilidad de Cassandra es horizontal, a diferencia de las bases de datos relacionales. La consistencia puede ser configurado según el dominio del negocio, puede ser una *escala distribuida*, *escala lineal*, *escala horizontal*.

La Escala Distribuida refiere a la repartición de la información está repartida a lo largo de los nodos del clúster. La ventaja que provee esta organización es la alta disponibilidad, así si algún nodo se cae el servicio no se degradará [3]. (Ver imagen 5)

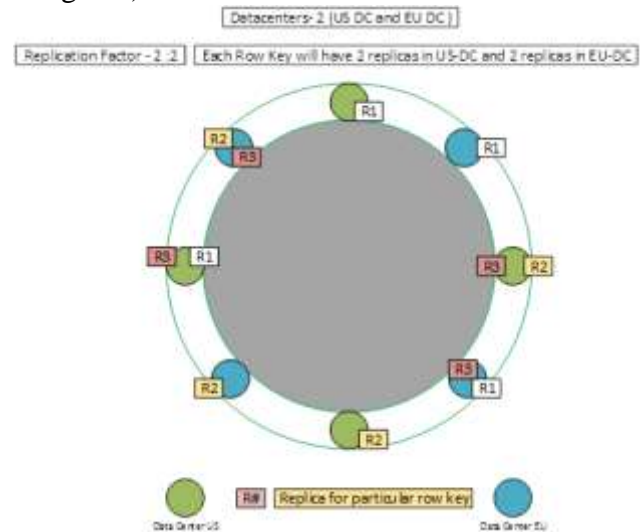


Imagen 5: Escala distribuida.

La Escala Lineal refiere a un rendimiento de forma lineal respecto al número de nodos que se inserten. Un caso práctico, radica en 300.000 operaciones serán soportadas por 4 nodos, a comparación de 150.000 operaciones/segundo en 3 nodos. Esta ventaja de medición de rendimiento, permite una planificación más apegada a la realidad, por tal en gastos temporales, espaciales y de dinero [3]. (Ver imagen 6)



Imagen 6: Escala lineal.

La Escala Horizontal refiere al crecimiento de los sistemas informáticos insertando más nodos en equipos hardware de coste mínimo. Aporta mediante una arquitectura Peer-ti-Peer, donde todos los nodos tienen la misma importancia, evitando la arquitectura master-Slave, democratizando los datos. Así cualquier nodo puede ser coordinador de una consulta. Subrogando al controlador el poder de elección

con que nodo sea el coordinador [3]. (Ver imagen 7)



Imagen 7: Escala horizontal.

III. METODOLOGÍA

El código fuente de esta práctica se encuentra en el archivo adjunto, con extensión “. cal”. Esta práctica se realizó en línea de comando.

A. Sentencias CRUD

Las sentencias desarrolladas son la creación, consulta, modificación y eliminación de los registros. El lenguaje de consulta es CQL (Cassandra Query Language)[8], el cual es similar al SQL[9] (Estructurada Query Language).

Creación de registro

La creación implica la identificación de las claves donde insertar los valores. La creación, así como las demás operaciones son similares al lenguaje SQL. Ver imagen 8.

```
operaciones-crud.cql
1
2
3 -- Insertar registro
4 INSERT INTO curso_big_data(en, nombre, carrera, tel_mov, email)
5 VALUES(001, 'Freddy', 'Ing Sistemas', '0988596203', 'freddy.abadl@ucuenca.edu.ec');
```

Imagen 8: Creación de nodo y relación

Consulta de registro

La consulta implica la selección de los datos dado una clave principal. La consulta, así como las demás operaciones son similares al lenguaje SQL. Ver imagen 9.

```
operaciones-crud.cql
10
11 -- Leer registro
12 SELECT nombre, email FROM curso_big_data;
```

Imagen 9: Consulta de nodo y relación

Actualización de registro

La actualización refiere a los valores directamente consultados por una clave principal. La actualización, así como las demás operaciones son similares al lenguaje SQL. Ver imagen 10.

```
operaciones-crud.cql
6
7 -- Actualizar registro
8 UPDATE curso_big_data SET email='freddy.abadl@ucuenca.edu.ec'
9 WHERE en=002;
```

Imagen 10: Actualización de registro

Eliminación de registro

La eliminación implica el borrado del valor dado una clave dada, y los valores asociadas a esta. La eliminación, así como las demás operaciones son similares al lenguaje SQL. Ver imagen 11.

```
operaciones-crud.cql
13 -- Eliminar registro
14 DELETE tel_mov FROM curso_big_data WHERE en=003;
```

Imagen 11: Eliminación de nodo y relaciones asociadas

IV. CONCLUSIONES

En esta práctica se pudo identificar los distritos Query permitidos en Cassandra para un CRUD, su implementación como representación de una base de datos clave valor, permitió identificar las ventajas de su uso. El no contar con un esquema definido, tal como lo hacen las bases de datos relacionales, permite implementar en problemas complejos, gracias a su flexibilidad. Uno de las ventajas que tiene Cassandra es el manejar con bases de datos pesadas. Los puntos que a futuro se debería estudiar es el uso en bases de datos pesadas y recurrentes, desde el punto de vista del desarrollador, así se podrá obtener un criterio más formado respecto a estas bases de datos NoSQL.

V. BIBLIOGRAFÍA

- [1] GrapherEverywhere, “Bases de Datos NoSQL | Qué son, marcas, tipos y ventajas,” 2019. <https://www.grapherverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/> (accessed Dec. 23, 2020).
- [2] M. Editors, “What is NoSQL? NoSQL Databases Explained | MongoDB,” 2020. <https://www.mongodb.com/nosql-explained> (accessed Dec. 23, 2020).
- [3] J. López, “Cassandra_Nosql,” p. 116, 2018.
- [4] P. V. Damian, “¿Qué son las bases de datos?,” 2007. <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/> (accessed Dec. 23, 2020).
- [5] EssentialSQL editors, “SQL ACID Database Properties Explained - Essential SQL,” 2015. <https://www.essentialsql.com/sql-acid-database-properties-explained/> (accessed Dec. 23, 2020).
- [6] AWS Editors, “¿Qué es una base de datos clave-valor?,” 2019. <https://aws.amazon.com/es/nosql/key-value/> (accessed Dec. 23, 2020).
- [7] A. Editors, “Bases de datos no relacionales | Bases de datos de gráficos | AWS,” 2019.

- <https://aws.amazon.com/es/nosql/> (accessed Dec. 23, 2020).
- [8] Apache Editors, “Documentation,” 2017. <https://cassandra.apache.org/doc/latest/cql/> (accessed Dec. 23, 2020).
- [9] TechTarget Editor, “¿Qué es SQL o lenguaje de consultas estructuradas? - Definición en WhatIs.com,” 2015. <https://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas> (accessed Dec. 23, 2020).