ALMACENAMIENTO DE DATOS

SHARDING

Daniel Blanco Calviño

SHARDING

- También llamado **particionamiento de datos.**
  - Se divide una gran BBDD en pequeñas partes más fáciles de manejar, shards.

- Clave de sharding o **partition key.**
  - Una o más columnas de una tabla que determinan cómo se distribuyen los datos.

EJEMPLO SHARDING

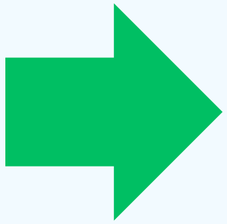
user (sin sharding)

id	username
1	daniel
2	noelia
3	lucia
4	diego
5	alba
6	mateo
7	victor
8	luis
9	valentina
10	sofia
11	silvia
12	manuel

EJEMPLO SHARDING

user (sin sharding)

id	username
1	daniel
2	noelia
3	lucia
4	diego
5	alba
6	mateo
7	victor
8	luis
9	valentina
10	sofia
11	silvia
12	manuel



user (shard 1)

id	username
1	daniel
4	diego
7	victor
10	sofia

user (shard 2)

id	username
2	noelia
5	alba
8	luis
11	silvia

user (shard 3)

id	username
3	lucia
6	mateo
9	valentina
12	manuel

ESTRATEGIAS SHARDING

- Sharding basado en **rango.**
  - Se almacena en cada shard los datos generados en un rango temporal.
  - Muy **eficiente para consultas de rango.**
  - **Desequilibrio en las cargas** de cada shard.

ESTRATEGIAS SHARDING

- Sharding basado en **rango.**
  - Se almacena en cada shard los datos generados en un rango temporal.
  - Muy **eficiente para consultas de rango.**
  - **Desequilibrio en las cargas** de cada shard.

- Sharding basado en **hashing.**
  - Se aplica función hash a la clave y se asigna a un shard en función del resultado.
  - Distribuye uniformemente la carga.
  - Ineficiente para consultas por rango. Los datos se encuentran en shards distintos.

ESTRATEGIAS SHARDING

- Sharding basado en **listas predefinidas.**
  - Se establece una lista de valores predefinidos para cada shard. Asignación manual.
  - Nos ofrece un **control granular** sobre la información que se distribuye a cada shard.
  - **Desequilibrio en las cargas** de cada shard si las listas no están bien diseñadas.

ESTRATEGIAS SHARDING

- Sharding basado en **listas predefinidas.**
  - Se establece una lista de valores predefinidos para cada shard. Asignación manual.
  - Nos ofrece un **control granular** sobre la información que se distribuye a cada shard.
  - **Desequilibrio en las cargas** de cada shard si las listas no están bien diseñadas.

- Sharding **round robin.**
  - Se inserta el primer elemento en el primer shard, el segundo en el segundo etc.
  - Algoritmo muy simple. Además cada shard tendrá un **número similar de datos.**
  - **Problema al localizar los datos.**
    - No depende de la clave de particionado.
    - Debemos mantener una estructura auxiliar que mapee cada dato al shard asignado.

DESAFÍOS AL IMPLEMENTAR SHARDING

- **Resharding** (redistribución de los datos).
  - Si añadimos / eliminamos un shard es probable que tengamos que redistribuir la información.
  - Se puede **mitigar**.
    - Selección de una estrategia que no distribuya en base al número de shards.
    - **Consistent Hashing.** Algoritmo de distribución de carga que aborda el problema.

DESAFÍOS AL IMPLEMENTAR SHARDING

- **Resharding** (redistribución de los datos).
  - Si añadimos / eliminamos un shard es probable que tengamos que redistribuir la información.
  - Se puede **mitigar**.
    - Selección de una estrategia que no distribuya en base al número de shards.
    - **Consistent Hashing.** Algoritmo de distribución de carga que aborda el problema.

- **Celebrity / hotspot problem.**
  - Algunos registros pueden ser accedidos con una mayor frecuencia.
    - Si muchos están en el mismo shard, este soportará una carga mucho mayor.
  - Es probable que se necesiten estrategias de shard más creativas (basadas en popularidad).

DESAFÍOS AL IMPLEMENTAR SHARDING

- Realizar **consultas** sobre la información.
  - Los datos se distribuyen en distintos shards.
    - Debemos **obtener información de distintos nodos y agregarla** para devolver el resultado final.
  - Problema transparente para nosotros. Se encarga el gestor que utilicemos.

SHARDING + REPLICACIÓN DE DATOS