Distribución y replicación

Distribución y replicación son los conceptos básicos que emplea Cassandra para garantizar la disponibilidad y tolerancia a fallos. En Cassandra ambas tareas se realizan do forma simultánea. Cuando un dato se distribuye, también se replica. Además de esto los datos se van a organizar en función de su clave primaria (PK). La PK (***Partition Key* no confundir con *primary key***) determina en qué nodo se van a escribir lo datos.

Elementos involucrados en la distribución y replicación

- **Nodos virtuales (*Vnodes*)**: aumentan el grado de granularidad de los datos.
- **Particionador**: determina en qué nodo se va a almacenar un dato en función de su PK.
- **Estrategia de replicación**: determina el número de copias que se van a almacenar de cada dato.
- ***Snitch***: determina la topología de la red.

Nodos virtuales (Vnodes)

Aumentan el grado de granularidad de los datos. Al comportarse como nodos *reales* permiten que un nodo almacene más datos que los que le corresponderían en una distribución sin *Vnodes*. Esto hace que haya datos replicados en más nodos y reduce el riesgo de que la caída de un nodo provoque la pérdida de datos.

El intervalo de PKs que le correspondería a un nodo se calcula a partir del valor de dos tokens. El primer token se calcula a partir del hash de la dirección IP del nodo. El segundo token se calcula a partir del hash del nombre del cluster.

Cuando añadimos un nuevo nodo, éste asume la responsabilidad sobre un conjunto de datos que le correspondía a otros nodos. Esto hace que se tenga que realizar un proceso de redistribución de datos. Este proceso se realiza de forma automática y transparente para el usuario.

La proporción de *Vnodes* por nodo es configurable.

Particionador

El particionador es el encargado de determinar en qué nodo se va a almacenar un dato en función de su PK. El algoritmo de distribución de datos utiliza una **función hash** para calcular el token de un dato a partir de su **PK**. El token es un número de 64 bits que se utiliza para determinar en qué nodo se va a almacenar el dato (en relación a los tokens del nodo). Como dijimos cada nodo tendrá dos tokens y el dado ha de almacenarse en el nodo *entre cuyos tokens* se encuentre el token del dato. Es decir, si un nodo tiene los tokens 1 y 100 y el hash del dato fuese 50 el dato se almacenará en dicho nodo.

Obviamente los tokens se generan de forma que todo el rango de posibles valores de hash de un dato esté cubierto por los rangos de tokens de los nodos.

Cassandra proporciona tres particionadores:

- Murmur 3Partitioner (por defecto).
- Random Partitioner.

- ByteOrdered Partitioner.

Todos ellos garantizan que los datos se distribuyen de forma uniforme entre los nodos.

Estrategia de replicación

Cassandra utiliza la replicación para asegurar la disponibilidad y tolerancia a fallos. El **factor de replicación** es el valor que indica el número de copias que se van a almacenar de cada dato y **no debe sobrepasar el número de nodos del *datacenter***. El valor de esta propiedad se puede modificar en tiempo de ejecución.

La replicación se realiza de forma automática y transparente para el usuario. Cassandra proporciona dos estrategias de replicación:

- SimpleStrategy (por defecto): Usado para clusters con un único *datacenter*. Las réplicas se distribuyen en los nodos de forma secuencial.
- NetworkTopologyStrategy: Usado para clusters con varios datacenters. Las réplicas se distribuyen en los nodos de forma secuencial en función de los datacenters. Se puede definir un factor de replicación distinto para cada datacenter.

Snitch

El snitch es el encargado de determinar la topología de la red. Es decir, determina a qué datacenter y a qué rack pertenece cada nodo. Cassandra proporciona varios snitches:

- Dynamic.
- GoogleCloudSnitch.
- Simple.
- Ec2Snitch.
- Rackinferring.
- ...