Primary key generation in sharded databased:

1. Local sequence: each shard server manage its own sequence

Problem: sequence conflict:

Eg. Insert 1-> shard 1: local seq.nextval = 1. Insert 2 -> shard 2: local seq.nextval = 1 conflict!

Solution: each shard server’s sequence starts with a different value. Step value of sequence = number of shard. However, if more shard servers are added, step value has to be re-adjusted.

Problem: master slave replication:

Master’s sequence value has to be in sync with slave’s sequence value. Sequence value replication may incur delay.

If master server goes down, slave’s sequence value has to continue from the latest value of master, and sequence value has to go in the reverse direction.

Extreme case: next sequence value is allocated from master, master goes down. Insert statement is performed at slave and sequence value is not yet replicated.

1. Global sequence: UUID

Pros: Global uniqueness is guaranteed.

Cons: UUID is too long. It is a random number that does not reflect insertion order. Storage cost and indexing performance will be issue

Snowflake is a better algorithm to generate globally unique ID

1. Global sequence: sequence server

All sequence values are allocated from a sequence server

Pros: Global uniqueness is guaranteed.

Cons: single point of failure, performance bottleneck

1. Global sequence: sequence server cluster

Multiple sequence server

Pros: avoid single server bottleneck. When one sequence server is down, sequence can be allocated from another server and it guarantees that there is no conflict, although sequence value may be discontinuous

Cons: adding new server requires manual work, such as adjust step and initial value of each existing sequence server.

MyCat way:

Use global sequence

File based: sequence\_conf.properties

Use local cache to avoid “nextval” request hitting sequence server each time.

Implementation:

If read/write separation is used, sequence retrieval query should be redirect to write host, although it is a “select” query

<http://blog.csdn.net/bluishglc/article/details/7710738>

<http://blog.csdn.net/freewebsys/article/details/44399901>

MASTER SLAVE REPLICATION

<http://hedengcheng.com/?p=892>