

Scalable Unique ID Generator (Primary Key)- Twitter Snowflake

Description

Introduction

While working with databases we come across something called as primary key, and one of the important rule is that primary key should be always unique for each newly created record. People who are new to the databases might think what's the big deal about generating unique ID's or primary keys. Simply we can start the first record with ID as 1 and then increment the ID up to the number we want. Traditional databases have a feature called *auto_increment* which will take care of incrementing the ID. But friends, I want to disappoint you by saying that this strategy will not work in distributed environment where we have multiple applications and databases running. *auto_increment* will work fine when we have a single database server setup.

Not only databases primary keys, but sometimes we would need unique ids for user ID or may be a bank account number. In this blog post I will discuss two possible solutions on how to generate unique id or to be precise how to generate a primary key.

UUID (Universally unique Identifier)

For the folks who don't know about UUID's, they are unique identifiers which have a very low chance that it will get duplicated. UUID can also be used in distributed applications without worrying about the ids getting duplicated. Here is how a sample UUID looks like **123e4567-e89b-12d3-a456-426655440000**. UUID consists of both alphabets and numbers. This approach would only be acceptable if your requirement allows the usage of alphanumeric in primary key.

Advantages

- UUID strategy can be used irrespective of the number of application or database servers you have.
- Mostly used as value for Correlation-ID's passed in request headers during REST service call. The purpose is to track the http request to see how many services the call went through and where the issue occurred if any failure.
- Scalability is not an issue.

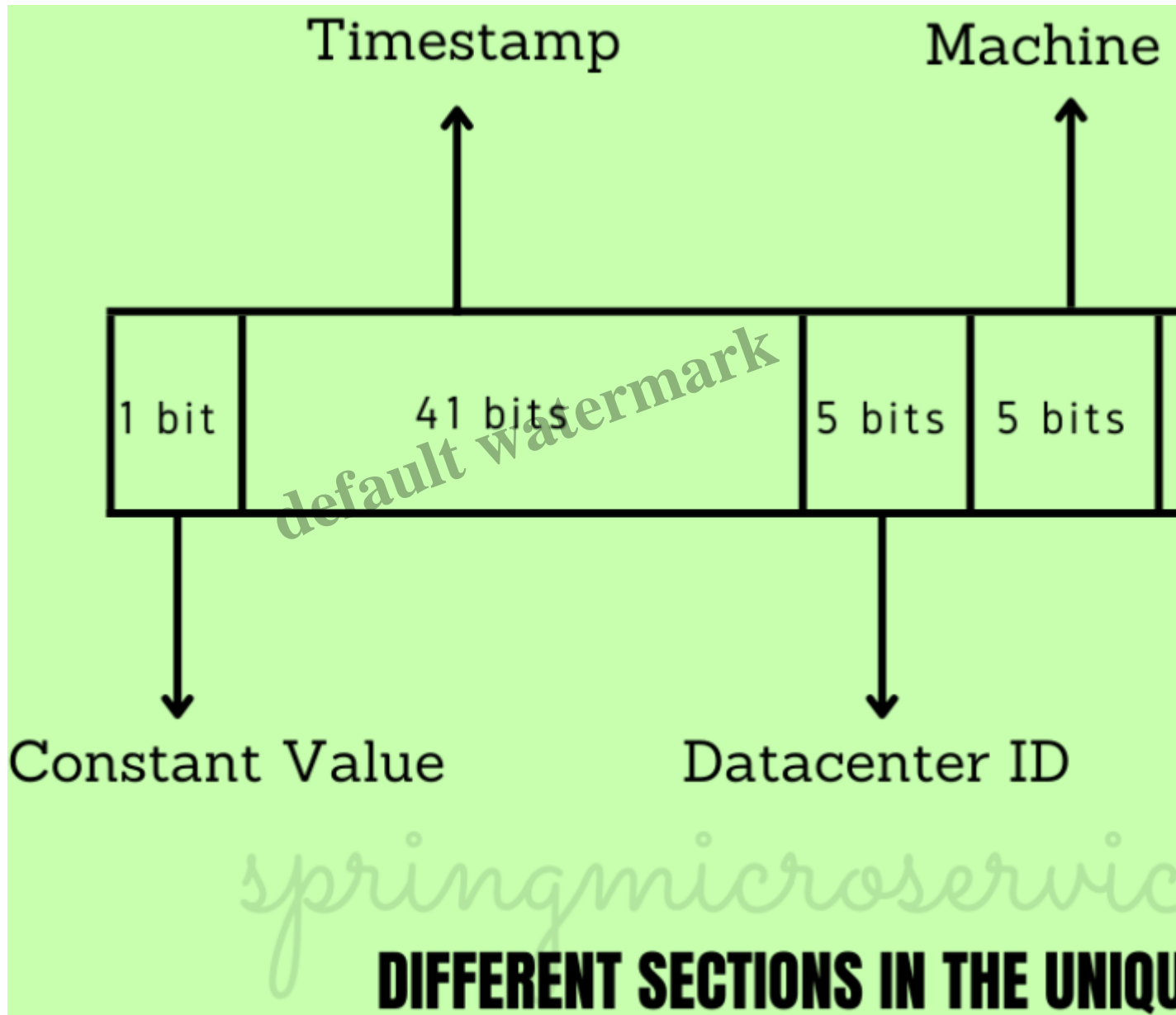
Disadvantages

- Sorting the records is difficult as the unique primary key contains alphabets also.
- It is 128 bit long.

Twitter Snowflake for ID generation

Snowflake is the ID generation strategy used by Twitter for their unique Tweet IDs. This particular Id generation strategy has been open sourced by Twitter.

ID generated using twitter snowflake method has various sections and each section has its own logic.



Constant Value – In the first section we usually have a constant value will can be 0.

Timestamp – In the timestamp section we append milliseconds since the epoch time. Epoch time is equivalent to some predefined date & time. So the timestamp will contain milliseconds calculated since the epoch time.

Datacenter Id – It is the value where the record will get stored.

Machine Id – It is the machine where the record is stored.

Sequence number – Sequence number gives a total of 4096 combination i.e. $2^{12} = 4096$. This field will usually be 0, unless we are generating more than one ID in a millisecond on the same server. A particular machine supports up to 4096 unique IDs per millisecond. The Sequence number is reset to 0 every millisecond.

I hope you guys liked this blog post on generating unique ids. Feel free to comment. ðŸ˜Š

Category

1. Design

Date Created

August 11, 2022

Author

kk-ravi144gmail-com

default watermark